STR2RTS: Refactored StreamIT benchmarks into statically analyzable parallel benchmarks for WCET estimation & real-time scheduling

Benjamin Rouxel
benjamin.rouxel@irisa.fr

Isabelle Puaut
isabelle.puaut@irisa.fr

Institut de Recherche en Informatique et Systèmes Aléatoires

Context

Multi-core embedded real-time systems

Scheduling/mapping applications on target architecture

Need to evaluate new scheduling/mapping techniques

Scheduling algorithms evaluation

- ➢ Schedulability analysis

  Use mathematical formulas to prove schedulability

- ➢ Implementation-based
  - Virtual: Simulate the schedule's result from algorithm
  - Real: Implement the scheduling algorithm on a platform

Task-set required at any evaluation level

High-level requirements

- Unbiased task-set
- Representative case
- Must be reproducible/reusable by others
- Independent from particular runtime/architecture

Task model properties

- Dependent and/or Independent
- Periodic and/or Sporadic
- Single-/Multi-graph
- Mixed-criticality

Task-level properties

- Worst-Case Execution Time (WCET)
  (source code, loop bounds, ...)
- Memory demand
- Periodicity
- Deadline
- Precedence relationship
- ...

Usable Existing Solutions Summary

	Synthetic	Real application	
Independent	Uunifast [1]	Task-set generator from TACleBench [3]	Periodic
	Uunifast [1]	Mälardalen [4] combination	Sporadic
Dependent	TGFF [2]	Debie [5], Papabench [6], Rosace [7]	Single-graph
	TGFF [2]		Multi-graph

Other existing benchmarks suites

- ➢ No design for multi-core
- ➢ No structured C source code
- ➢ No identified Tasks and dependency
- ➢ Under licence

SPEC CPU 2006, PolyBench, ParMiBench, JemBench, ParaSuite, StreamIT, Parsec, UTDSP, ...

TACleBench suite: not enough parallel benchmarks

STR2RTS:
STReamit to Real Time System

StreamIT background

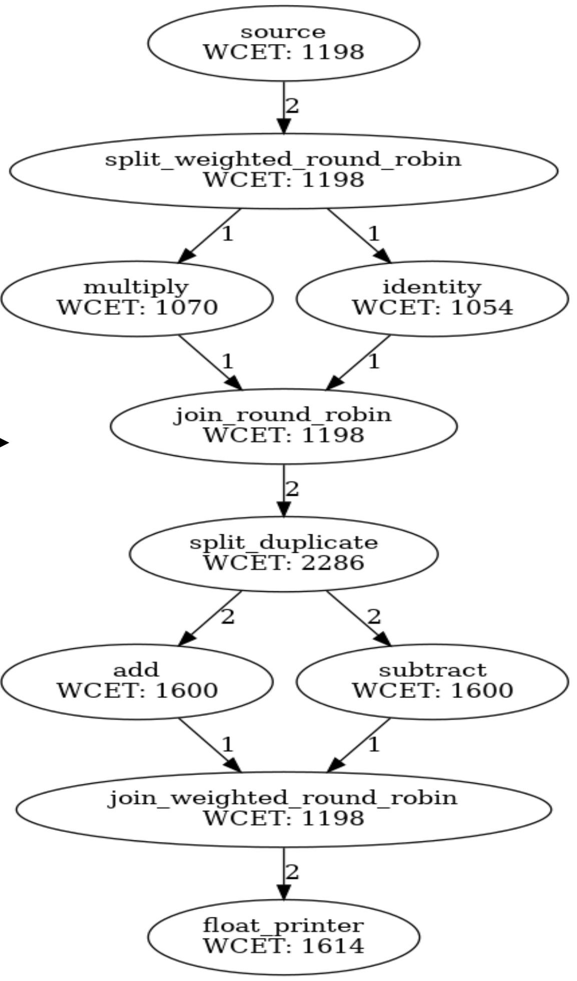
Programming language & compilation infrastructure
Benchmark suite

Static structure of streaming applications
    graph of filters (fork-join graph)
Fixed data-rate known at compile time
Multi-level exploitable parallelism
    data, task, iteration parallelism (pipelining)

Example of StreamIt benchmark: radix-2 case of a Fast Fourier Transform

```
void->void pipeline FFT4 {
    add OneSource();
    add FFTKernel(2);
    add FloatPrinter();
}
float->float pipeline FFTKernel (int N) {
    for (int i=1; i<N; i*=2) {
        add Butterfly(i, N);
    }
}
float->float pipeline Butterfly (int N, int W) {
    add splitjoin {
        split roundrobin(N, N);
        add Identity<float>();
        add Multiply();
        join roundrobin();
    };
    add splitjoin {
        split duplicate;
        add Add();
        add Subtract();
        join roundrobin(N, N);
    };
}
void->float filter OneSource {
    work push 1 { push(1); }
}
float->float filter Multiply {
    work push 1 pop 1 { push(2 * pop()); }
}
float->float filter Add {
    work push 1 pop 2 {
        push(peek(0) + peek(1));
        pop();
        pop();
    }
}
float->float filter Subtract {
    work push 1 pop 2 {
        push(peek(0) - peek(1));
        pop();
        pop();
    }
}|
float->void filter FloatPrinter {
    work pop 1 { println(pop()); }
}
```

$\longrightarrow$



What we propose

Dependent tasks represented by a single-graph

a description of the task's dependency (XML file)

Every information to compute a static WCET is available

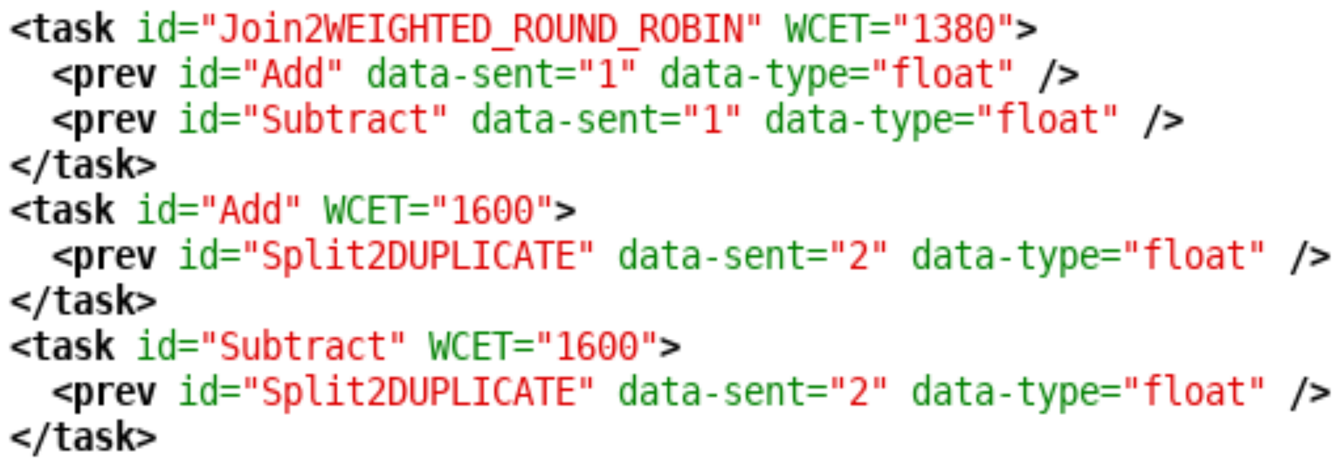
structured C source code, loop bounds (C source file)

Example of STR2RTS benchmark: radix-2 case of a Fast Fourier Transform

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<appl>
  <tasks>
  <task id="Split2DUPLICATE" WCET="2286">
    <prev id="Join1ROUND_ROBIN" data-sent="2" data-type="float" />
  </task>
  <task id="Join2WEIGHTED_ROUND_ROBIN" WCET="1380">
    <prev id="Add" data-sent="1" data-type="float" />
    <prev id="Subtract" data-sent="1" data-type="float" />
  </task>
  <task id="Add" WCET="1600">
    <prev id="Split2DUPLICATE" data-sent="2" data-type="float" />
  </task>
  <task id="Subtract" WCET="1600">
    <prev id="Split2DUPLICATE" data-sent="2" data-type="float" />
  </task>
  <task id="FloatPrinter" WCET="1614">
    <prev id="Join2WEIGHTED_ROUND_ROBIN" data-sent="2" data-type="float" />
  </task>
  <task id="OneSource" WCET="1198"></task>
  <task id="Split1WEIGHTED_ROUND_ROBIN" WCET="1380">
    <prev id="OneSource" data-sent="2" data-type="float" />
  </task>
  <task id="Join1ROUND_ROBIN" WCET="1198">
    <prev id="Identity" data-sent="1" data-type="float" />
    <prev id="Multiply" data-sent="1" data-type="float" />
  </task>
  <task id="Identity" WCET="1054">
    <prev id="Split1WEIGHTED_ROUND_ROBIN" data-sent="1" data-type="float" />
  </task>
  <task id="Multiply" WCET="1070">
    <prev id="Split1WEIGHTED_ROUND_ROBIN" data-sent="1" data-type="float" />
  </task>
</tasks>
<processors>
    <!-- processor list parameters -->
</processors>
<config>
    <!--  scheduler configuration -->
</config>
</appl>
```

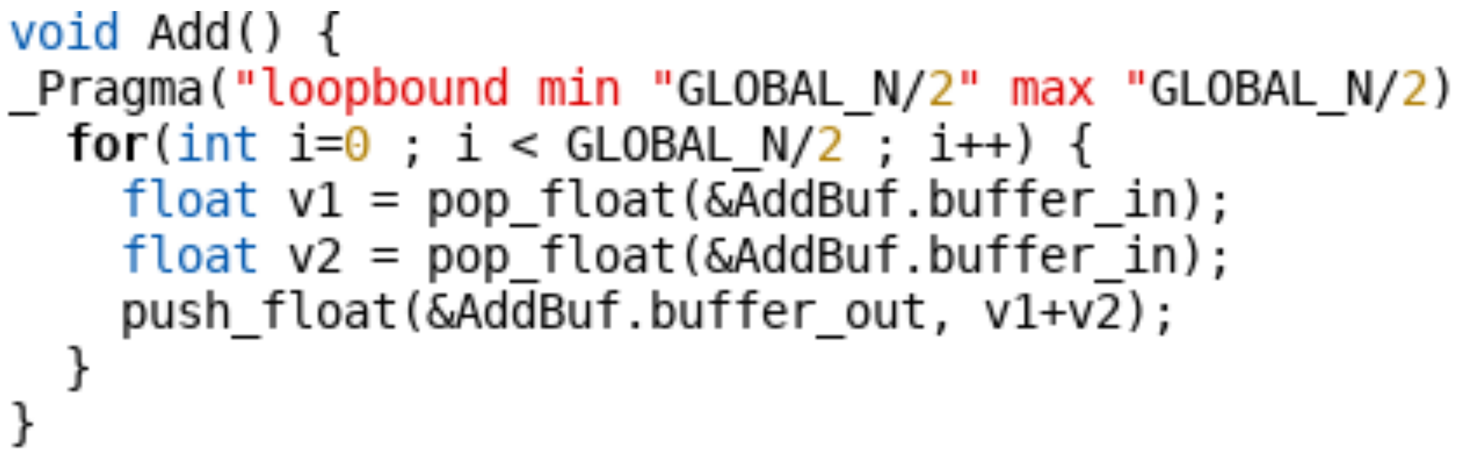
```c
#include "FFT4.h"

void fft4_one_source() { ... }
void fft4_identity() { ... }
void fft4_multiply() { ... }
void fft4_add() {
_Pragma("loopbound min "GLOBAL_N/2" max "GLOBAL_N/2)
  for(int i=0 ; i < GLOBAL_N/2 ; i++) {
    float v1 = pop_float(&AddBuf.buffer_in);
    float v2 = pop_float(&AddBuf.buffer_in);
    push_float(&AddBuf.buffer_out, v1+v2);
  }
}
void fft4_subtract() { ... }
void fft4_float_printer() { ... }
void fft4_init() { ... }
void fft4_split1_weighted_round_robin( uint32_t nb ) { ... }
void fft4_join1_round_robin() { ... }
void fft4_split2_duplicate() { ... }
void fft4_join2_weighted_round_robin( uint32_t nb ) { ... }
int sequential_main( int argv, char **argc ) {
  fft4_init(); |\label{lst:ex-c:ini}|
_Pragma("loopbound min "MAX_ITERATION" max "MAX_ITERATION)
  for( int i=0 ; i < MAX_ITERATION ; i++ ) {
    fft4_OneSource();
_Pragma("loopbound min "(GLOBAL_N/2-1)" max "(GLOBAL_N/2-1))
    for( int j = 1 ; j < GLOBAL_N ; j *= 2 ) {
      fft4_split1_weighted_round_robinv(j);
        fft4_identity();
        fft4_multiply();
      fft4_join1_round_robin();
      fft4_split2_duplicate();
        fft4_add();
        fft4_subtract();
      fft4_join2_weighted_round_robin(j);
    }
    fft4_float_printer();
  } |\label{lst:ex-c:m:lo}|
  return EXIT_SUCCESS;
} |
```

Example of STR2RTS benchmark: radix-2 case of a Fast Fourier Transform



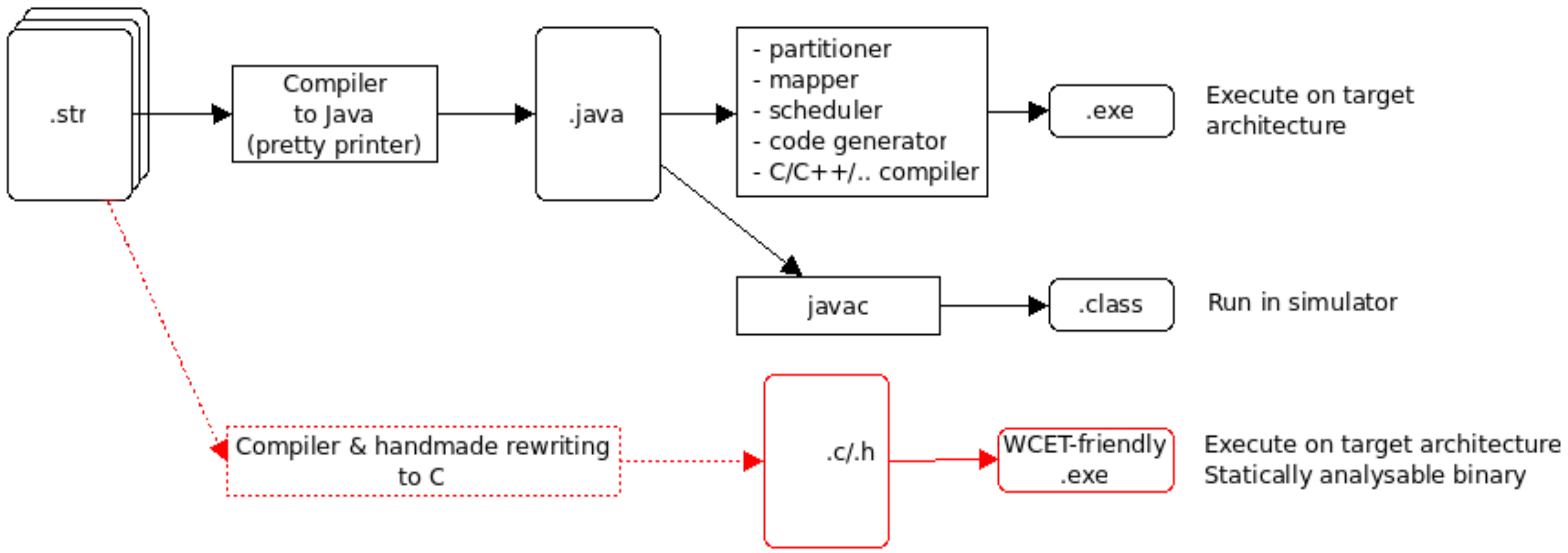
```
<task id="Join2WEIGHTED_ROUND_ROBIN" WCET="1380">
  <prev id="Add" data-sent="1" data-type="float" />
  <prev id="Subtract" data-sent="1" data-type="float" />
</task>
<task id="Add" WCET="1600">
  <prev id="Split2DUPLICATE" data-sent="2" data-type="float" />
</task>
<task id="Subtract" WCET="1600">
  <prev id="Split2DUPLICATE" data-sent="2" data-type="float" />
</task>
```

Example of STR2RTS benchmark: radix-2 case of a Fast Fourier Transform



```
void Add() {
_Pragma("loopbound min "GLOBAL_N/2" max "GLOBAL_N/2)
  for(int i=0 ; i < GLOBAL_N/2 ; i++) {
    float v1 = pop_float(&AddBuf.buffer_in);
    float v2 = pop_float(&AddBuf.buffer_in);
    push_float(&AddBuf.buffer_out, v1+v2);
  }
}
```

Exploiting the StreamIT infrastructure


.str
Compiler to Java (pretty printer)
.java
- partitioner
- mapper
- scheduler
- code generator
- C/C++/.. compiler
.exe
Execute on target architecture
javac
.class
Run in simulator
Compiler & handmade rewriting to C
.c/.h
WCET-friendly .exe
Execute on target architecture
Statically analysable binary

Available benchmarks

11 benchmarks

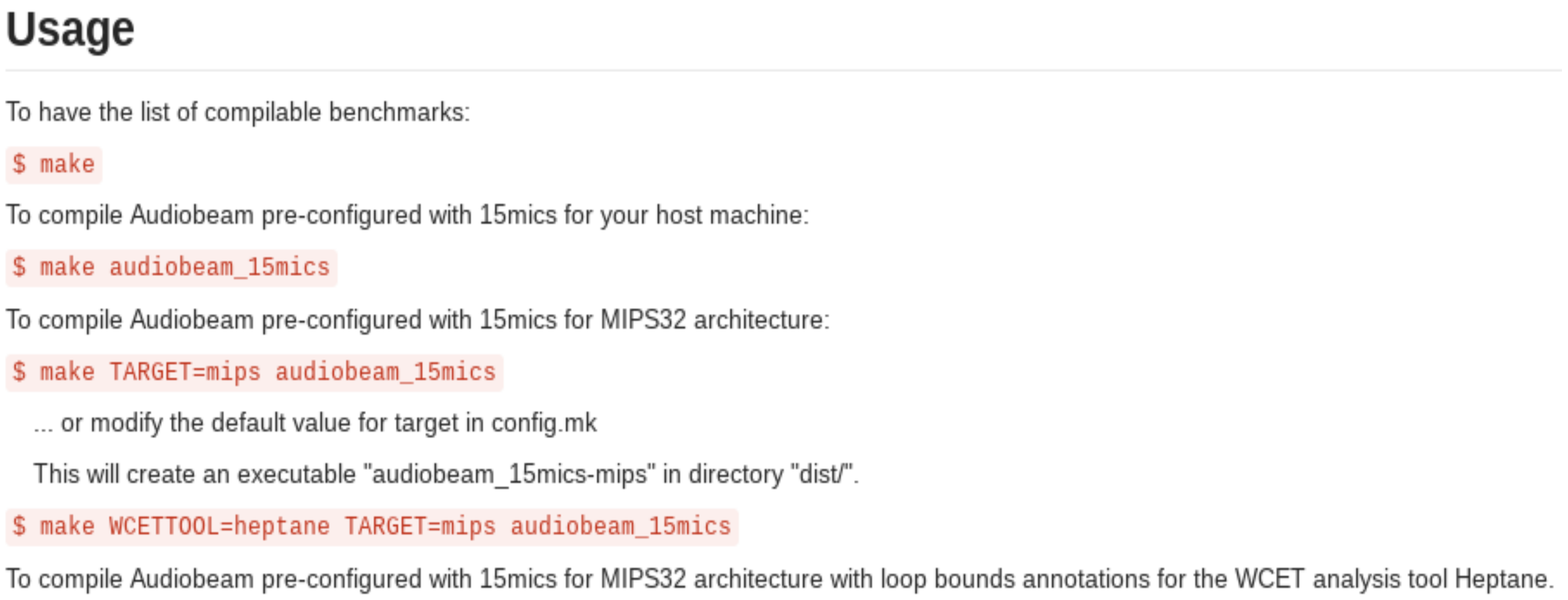
Name	#tasks	Width	WCET (cycles)	Data (tokens)	#Basic blocks
			<avg, standard deviation>		
802.11a	[119;132]	[7;18]	2.39e5, 6.35e5	596, 2238	1584
6/9/12/18/24/36					
Audiobeam	20	15	273, 1094	3, 5	386
Beamformer	56	12	1.25e4, 9.65e4	4.6, 10	459
CFAR	4	1	1.58e4, 1.55e4	288, 425	375
Complex-FIR	3	1	501, 655	1.3, 0	336
DCT2	40	16	1.69e4, 3958	57.6, 91	437
DES	423	8	2045, 1417	35.7, 29	621
FFT2	26	2	2.94e5, 3.03e5	137.8, 49	477
FFT4	42	2	1337, 450	23.6, 8	566
FilterBankNew	52	6	6315, 8306	8.9, 11	446
FMRadio	43	12	1632, 2234	1.6, 1	486

Properties of available benchmarks

Name	Use math library	Use I/O file	Two versions / code reuse
802.11a	yes	no	yes
Audiobeam	yes	yes	no
Beamformer	yes	no	no
CFAR	yes	no	no
Complex-FIR	no	yes	no
DCT2	yes	yes	no
DES	no	no	yes
FFT2	yes	no	no
FFT4	no	no	no
FilterBankNew	no	no	no
FMRadio	yes	no	no

How to

https://gitlab.inria.fr/brouxel/STR2RTS



Usage

To have the list of compilable benchmarks:

```
$ make
```

To compile Audiobeam pre-configured with 15mics for your host machine:

```
$ make audiobeam_15mics
```

To compile Audiobeam pre-configured with 15mics for MIPS32 architecture:

```
$ make TARGET=mips audiobeam_15mics
```

... or modify the default value for target in config.mk

This will create an executable "audiobeam_15mics-mips" in directory "dist/".

```
$ make WCETTOOL=heptane TARGET=mips audiobeam_15mics
```

To compile Audiobeam pre-configured with 15mics for MIPS32 architecture with loop bounds annotations for the WCET analysis tool Heptane.

Conclusion

Dependent tasks
Statically analyzable
XML description + structured C source
Inspired by StreamIT

https://gitlab.inria.fr/brouxel/STR2RTS

[1] Enrico Bini and Giorgio C Buttazzo. Measuring the performance of schedulability tests. Real-Time Systems, 2005.

[2] Robert P Dick, David L Rhodes, and Wayne Wolf. Tgff: task graphs for free. In 6th workshop on Hardware/software codesign, 1998.

[3] Yorick De Bock, Sebastian Altmeyer, Jan Broeckhove, and Peter Hellinckx. Task-set generator for schedulability analysis using the taclebench benchmark suite. Workshop : EWiLi 2016.

[4] Jan Gustafsson, Adam Betts, Andreas Ermedahl, and Björn Lisper. The Mälardalen WCET benchmarks – past, present and future. OCG 2010.

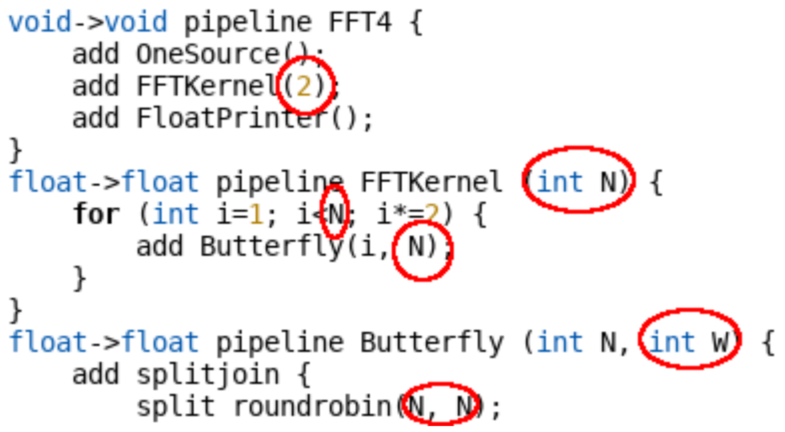
[5] Debie1. URL: https://www.irit.fr/wiki/doku.php?id=wtc:benchmarks:debie1.

[6] Fadia Nemer, Hugues Cassé, Pascal Sainrat, Jean-Paul Bahsoun, and Marianne De Michiel. Papabench: a free real-time benchmark. In OASIcs-OpenAccess Series in Informatics, volume 4, 2006.

[7] Claire Pagetti, David Saussié, Romain Gratia, Eric Noulard, and Pierre Siron. The ROSACE Case Study: From Simulink Specification to Multi/Many-Core Execution. In 20th IEEE RTAS 2014.

A necessary handmade step

- Hard transformation passes
- Loopbound identification
- Hard to debug compiler
- No major commit in *apps* folder on github for 6 years



```
void->void pipeline FFT4 {
    add OneSource();
    add FFTKernel(2);
    add FloatPrinter();
}
float->float pipeline FFTKernel (int N) {
    for (int i=1; i<N; i*=2) {
        add Butterfly(i, N);
    }
}
float->float pipeline Butterfly (int N, int W) {
    add splitjoin {
        split roundrobin(N, N);
```