TACLeBench: A Benchmark Collection to Support Worst-Case Execution Time Research

Heiko Falk and many more
Martin Schoeberl
Technical University of Denmark

The Need for Benchmarks

- Engineering based research uses experiments to evaluate ideas
- Best to have reproducible experiments
- Agree on a set of programs
- Open source for easy usage
- Versioning of TACLeBench for reproducibility

Research Targets

- WCET analysis
  - High level and low level
- Compiler optimization for WCET
- Hardware (processors) for real-time systems
- Task sets for scheduling

- Missing: measurement based WCET analysis
  - Maybe join the effort?

Why Not Using Other Benchmarks?

- Benchmarks for WCET analysis, compilation, architecture,…
  - Different needs than average case measurements
- Need to be WCET analyzable
  - Where are the loop bounds in MiBench and others?
- Shall be self contained
  - Assuming a bare metal runtime

Program Collection

- Started with Mälardalen benchmark collection
  - Which itself is a collection from different sources
- Additional programs collected by Heiko Falk
  - Part of the wcc research
- MiBench, DEBIE, PapaBench
- Additional application benchmarks
- Currently 54 benchmark programs

Program Types

- Kernel programs
  - Search, crc, matrix
  - 18 – 992 LoC
- Sequential programs
  - Mostly DSP programs
  - 117 – 2710 LoC
- Tests – to stress WCET tools
  - 35 - 4235 LoC

Program Types

- Parallel
  - Debie
    - 6615 LoC
  - PapaBench
    - 6336 LoC
- Application
  - Lift controller
    - 361 LoC
  - Power window
    - 2533 LoC

Properties

- Completely self contained
  - No libraries needed
    - If needed, source is added
  - No operating system needed
  - Nice for embedded bare metal systems
- Input data part of the source
- Easy to compile – no build tool magic ☺
  - cc/gcc/clang *.c

Issue with Original Source

- All input data fixed and part of the source
  - Practically single path code
- No return value
- Smart compilers can optimize it all away
- WCET analyzers can detect the single path
- Fixed with
  - Input data volatile
  - main return value dependent on computed results

Licenses

- We tracked down all possible licenses
- When not available benchmark dropped
- Some had even a license that disallowed distribution – dropped
- Now all is in place
  - No future need to investigate

Source Organization

- Unique identifiers
  - Prepend benchmark name
  - To enable combination of benchmarks
- Split into
  - Initialization
  - Computation
  - Result computation

Coding Style

- Adapted to common coding style
  - Line breaks, white spaces,…
- Naming of functions and global variables
- Function forward declaration
- Header comments on basic facts of the benchmark
  - What it does
  - Where it comes from

Example Code

```
void select_init( void );
int select_return( void );
int select_main( void );
int main( void );

int main( void )
{
  select_init();
  select_main();

  return( select_return() - 35 != 0 );
}
```

Source Annotations

- Heiko's wcc annotation
- Entry point and loop bounds
  - _Pragma ("entrypoint")
  - _Pragma("loopbound min 1 max 20")
- Supported by
  - wcc
  - clang for Patmos
  - Any other compiler/tool?

TACLeBench Usage

- Use a concrete version and document it
  - Currently it is V 1.9
    - Minor changes are pending
  - Will be V 2.0 at paper publication
- Use all benchmarks if possible
  - No selection for *best* representation of result
  - If really need to subset
    - Use one class or
    - Do a random selection

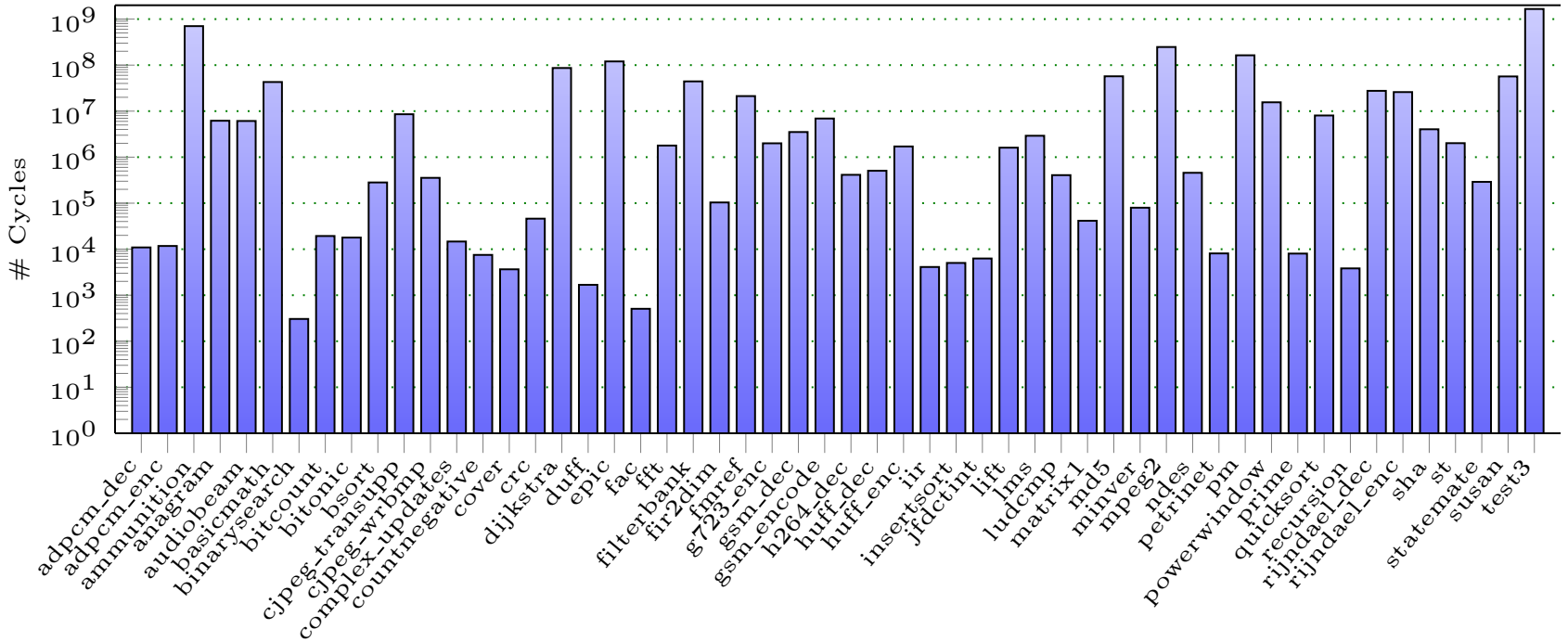
Sanity Checks

- Regression tests with online results
  - http://tacle.cs.fau.de/
- Compiled with gcc, g++, and clang
- Benchmarks execute on Patmos simulator
- Clang analyzer for program bugs

Future Development

- Git master head is a stabile version
- Development in development branch
- Merge to master with a concrete version
- Contributions are very welcome
  - Use git fork and pull request
  - We need more real world applications

Results


Cycles
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
10^9
adpcm_dec
adpcm_enc
ammunition
anagram
audiobeam
basicmath
binarysearch
bitcount
bitonic
bsort
cjpeg-transupp
cjpeg-wrbmp
complex_updates
countnegative
cover
crc
dijkstra
duff
epic
fac
fft
filterbank
fir2dim
fmref
g723_enc
gsm_dec
gsm_encode
h264_dec
huff_dec
huff_enc
iir
insertsort
jfdctint
lift
lms
ludcmp
matrix1
md5
minver
mpeg2
ndes
petrinet
pm
powerwindow
prime
quicksort
recursion
rijndael_dec
rijndael_enc
sha
st
statemate
susan
test3

The Link

- https://github.com/tacle/tacle-bench

Summary

- WCET research needs benchmarks
- TACLeBech is a program collection for WCET related research
- Collection of open source programs
- Fully annotated
- One program style
- With version control for reproducible results
- Paper is in the repository