WCC 2011


3rd Worst Case Challenge
WCC'11

aims and participants

tools and benchmarks

problems and results

experiences and conclusions

Presented by
Björn Lisper, Mälardalen University
Niklas Holsti, Tidorum Ltd
Authors

Reinhard von Hanxleden, Niklas Holst, Björn Lisper, Erhard Ploedereder, Reinhard Wilhelm (Eds.), Armelle Bonenfant, Hugues Cassé, Sven Bünte, Wolfgang Fellger, Sebastian Gepperth, Jan Gustafsson, Benedikt Huber, Nazrul Mohammad Islam, Daniel Kästner, Raimund Kirner, Laura Kovács, Felix Krause, Marianne de Michiel, Mads Christian Olesen, Adrian Prantl, Wolfgang Puffitsch, Christine Rochange, Martin Schoeberl, Simon Wegener, Michael Zolda, Jakob Zwirchmayr
WCET Challenge: Aims

- **Use** many WCET tools on the **same benchmark programs**
  - challenge tools with new benchmarks
  - enable cross-tool comparisons along several dimensions
  - demonstrate novel abilities of tools
  - present wide spectrum of tools to potential users

- **Comparison dimensions** include:
  - automatic analysis vs. manually written annotations
    - eg. loop bounds and infeasible paths
  - expressiveness and usability of annotation mechanism
    - eg. using source-code names vs. machine addresses
  - precision of WCET bounds or estimates
    - difficulty: true WCETs known only for small benchmarks

- **Both static analysis tools and measurement-based tools**
  - difficulty: few benchmarks have test suites
From 2006 through 2008 to 2011

- **First** Challenge 2006 – pioneering success
  - several WCET tools, benchmarks, processors
  - few comparable combinations of benchmark + processor

- **Second** Challenge 2008
  - one common target processor (system)
  - better definition of benchmarks, problems, and results
  - include pure flow-analysis problems and new benchmarks

- **Third** Challenge 2011
  - one simple and one complex common target processor
  - other processors allowed, eg. Java processor
  - benchmarks: some old, some new, some borrowed
  - experiment at Daimler automotive:
    - students use WCET tools on industrial code
    - gauge learning effort, usability, skills needed
  - see Wiki at www.mrtc.mdh.se/projects/WCC/2011/
WCC'11 process

benchmark sources
PapaBench: IRIT, debie1: SSF/Tidorum

benchmark code, analysis problems

WCC'11 Wiki

public

results

WCC'11 organizers and editors

report

tool developers

results, report contributions

www.mrtc.mdh.se/projects/WCC/2011/

Daimler

benchmark code, analysis problems

application knowledge

student analysts on Daimler site

tools, advice

WCET 2011, Porto, 2011-07-05
### Participants through the ages

<table>
<thead>
<tr>
<th>Tool</th>
<th>Source</th>
<th>2006</th>
<th>2008</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>aiT</td>
<td>AbsInt GmbH</td>
<td>SWb</td>
<td>SWb</td>
<td></td>
</tr>
<tr>
<td>Astrée</td>
<td>AbsInt GmbH</td>
<td></td>
<td></td>
<td>SFs</td>
</tr>
<tr>
<td>Bound-T</td>
<td>Tidorum Ltd</td>
<td>SWb</td>
<td>SWb</td>
<td>SWb</td>
</tr>
<tr>
<td>Chronos</td>
<td>National Univ. Singapore</td>
<td></td>
<td></td>
<td>SWsb</td>
</tr>
<tr>
<td>FORTAS</td>
<td>TU Vienna</td>
<td></td>
<td></td>
<td>Hsb</td>
</tr>
<tr>
<td>METAMOC</td>
<td>Aalborg University</td>
<td>(H)</td>
<td></td>
<td>(SWsb)</td>
</tr>
<tr>
<td>MTime</td>
<td>TU Vienna</td>
<td>(H)</td>
<td></td>
<td>H</td>
</tr>
<tr>
<td>oRange + OTAWA</td>
<td>IRIT</td>
<td>SWsb</td>
<td>SWsb</td>
<td></td>
</tr>
<tr>
<td>RapiTime</td>
<td>Rapita Systems Ltd</td>
<td></td>
<td></td>
<td>(Hsb)</td>
</tr>
<tr>
<td>SWEET</td>
<td>Mälardalen University</td>
<td>SWb</td>
<td></td>
<td>SFs</td>
</tr>
<tr>
<td>TimeWeaver</td>
<td>AbsInt GmbH</td>
<td></td>
<td></td>
<td>Hb</td>
</tr>
<tr>
<td>TuBound</td>
<td>TU Vienna</td>
<td></td>
<td></td>
<td>SWsb</td>
</tr>
<tr>
<td>WCA</td>
<td>TU Vienna, TU Denmark</td>
<td></td>
<td></td>
<td>SWc</td>
</tr>
<tr>
<td>wcc</td>
<td>Dortmund U Technology</td>
<td></td>
<td></td>
<td>SFsb</td>
</tr>
</tbody>
</table>

#### Tool Characteristics

<table>
<thead>
<tr>
<th>Tool Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S--</td>
<td>Static</td>
</tr>
<tr>
<td>H--</td>
<td>Hybrid</td>
</tr>
<tr>
<td>-W-</td>
<td>WCET &amp; flow</td>
</tr>
<tr>
<td>-F-</td>
<td>Flow only</td>
</tr>
<tr>
<td>(---)</td>
<td>No results</td>
</tr>
<tr>
<td>--s</td>
<td>Source</td>
</tr>
<tr>
<td>--c</td>
<td>Bytecode</td>
</tr>
<tr>
<td>--b</td>
<td>Binary</td>
</tr>
<tr>
<td>--sb</td>
<td>Source (flow) and binary (WCET)</td>
</tr>
</tbody>
</table>
## Target processors

### the “simple” common processor

<table>
<thead>
<tr>
<th>Processor</th>
<th>Tools</th>
<th>2006</th>
<th>2008</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM7</td>
<td>aiT, Bound-T, METAMOC, OTAWA, RapiTime</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>ARM9</td>
<td>SWEET</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C16x</td>
<td>aiT, TuBound</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ERC32 (SPARC)</td>
<td>Bound-T</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>JOP (Java)</td>
<td>WCA</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>MPC5553/5554</td>
<td>aiT, OTAWA</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>MPC565</td>
<td>aiT</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Renesas H8/300</td>
<td>Bound-T</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimpleScalar</td>
<td>Chronos</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriCore 1796</td>
<td>FORTAS</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

### the “complex” common processor(s)

<table>
<thead>
<tr>
<th>Number of tools:</th>
<th>N = 1</th>
<th>N = 2</th>
<th>N = 3 or more</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Benchmark</td>
<td>Source (immediate)</td>
<td>2006</td>
<td>2008</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>Mälardalen bm's</td>
<td>Mälardalen University</td>
<td>Os</td>
<td></td>
</tr>
<tr>
<td>PapaBench</td>
<td>TRACES/IRIT</td>
<td>Os</td>
<td>Os</td>
</tr>
<tr>
<td>rathijit bm's</td>
<td>Saarland University</td>
<td></td>
<td>Os</td>
</tr>
<tr>
<td>debie1</td>
<td>SSF/Tidorum Ltd</td>
<td></td>
<td>Rs</td>
</tr>
<tr>
<td>Daimler bm</td>
<td>Daimler</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For WCC'2011, all benchmarks are “real” programs:
- **PapaBench**: from Paparazzi Unmanned Aerial Vehicle controller
- **debie1**: from DEBIE-1 satellite-based scientific instrument
- **Daimler**: from truck control system
Benchmark details

- **PapaBench**
  - two programs (autopilot, fly-by-wire), originally 2 CPUs
  - C; 5020 lines, 1521 semicolons (“airborne” part)
  - single thread per program, cyclic scheduler + interrupts

- **debie1**
  - one program, originally six threads under preemptive RTK
  - C; 10228 lines, 1748 semicolons (excluding test harness)
  - test suite included (no I/O or RTK needed)

- **Daimler**
  - control system for trucks, eg. collision detection
  - C; size not revealed
  - parts analysed: an interrupt handler; an initialization routine; two calculation routines; one complete task
“Analysis Problem” definitions

- **Conditions + question**
- **Flow analysis** problems
  - how many times can X occur under these conditions?
  - eg. how many times can **Foo** call **Bar**?
- **WCET analysis** problems
  - give a WCET bound on Foo under these conditions
- “**Subprogram**”
  - the root procedure/function (or part of it, in some cases)
- “**Inputs**”
  - bounds on values of relevant input data
- “**Other constraints**”
  - other conditions on the scenarios/executions to be included
  - for example: no call of **Reboot** is executed
Example analysis problems

- debie1, problem 2a/F1 (flow analysis):
  - assuming that:
    
    \[
    \text{telemetry\_pointer} < \text{telemetry\_end\_pointer} \quad \text{and} \quad \text{telemetry\_pointer} \neq \&\text{telemetry\_data\_time}
    \]
  - how many times can \text{TM\_InterruptService} call \text{Send\_ISR\_Mail} ?

- PapaBench AutoPilot, problem A2a (WCET analysis):
  - assuming that:
    
    \[
    \text{pprz\_mode} = \text{PPRZ\_MODE\_HOME} \quad \text{(value 3)}
    \]
  - what is a WCET bound for \text{navigation\_update} ?

- Daimler benchmark:
  - analysis problems not shown on the Wiki
  - results give WCET of four functions
Questions and answers, overall

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Type of question</th>
<th>Number of questions</th>
<th>debie1 Flow</th>
<th>WCET</th>
<th>PapaBench Flow</th>
<th>WCET</th>
<th>Daimler WCET</th>
</tr>
</thead>
<tbody>
<tr>
<td>aiT</td>
<td></td>
<td></td>
<td>15</td>
<td>22</td>
<td>3</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>Astrée</td>
<td></td>
<td></td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bound-T</td>
<td></td>
<td></td>
<td>14</td>
<td>18</td>
<td>6</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>FORTAS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METAMOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OTAWA</td>
<td></td>
<td></td>
<td>15</td>
<td></td>
<td>5</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>SWEET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TimeWeaver</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TuBound</td>
<td></td>
<td></td>
<td>15</td>
<td>18</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>WCA</td>
<td></td>
<td></td>
<td>13</td>
<td></td>
<td>11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Number of posed and answered analysis problems in WCC’11.
Obstacles in the benchmarks

- **Floating point computations** (debie1, PapaBench)
  - loops in SW FP libraries difficult to bound automatically
    - annotations needed
  - few tools do value analysis of FP variables
    - loops with FP conditions must be annotated

- **Source code missing** for library functions (FP & others?)
  - source-code analyzers (Astrée, oRange) need it
  - manual annotation easier if source code is available

- **MPC (PowerPC)** instruction set subsets
  - debie1 code contains Variable Length Encoding instructions
    - VLE was not supported yet in OTAWA

- Infinite loops (eg. Daimler benchmark “task” code)

- Large number of tests in debie1, giving large traces
  - hampered TimeWeaver
Obstacles in the tools

- MPC instruction set is complex, has subsets
  - debie1 code contains Variable Length Encoding instructions
    - VLE was not supported yet in OTAWA
- TuBound failed to install at Daimler
  - depends on other SW, some with licensing restrictions
- SWEET did not attempt Daimler benchmark
  - C-to-ALF translator hard to install, many dependencies
- TimeWeaver had problems with tracing hardware
- New or prototype tools, still in development:
  - FORTAS
  - METAMOC
Tool developments during WCC'11

- aiT: increase automation, reduce annotations
- Astrée: streamlined export of flow information
- METAMOC: “gained many improvements”
- OTAWA: began extension to full MPC instruction set
- SWEET: new treatment of data at absolute address
- Tidorum (Bound-T): new tools for ET measurement
- TimeWeaver: handle incomplete traces, multi-exit funcs
- WCA: improved annotation language
Benchmark ports during WCC'11

- Done in order to enable participation in WCC'11
- PapaBench
  - port to TriCore 1796 (FORTAS team)
  - port to C167 (TuBound team)
  - translation into ALF (SWEET team)
  - Java version improved for JOP (WCA team)
- debie1
  - port to C167 (TuBound team)
  - port to MPC5554 (Simon Wegener, AbsInt)
  - translation to Java / JOP (WCA team)
- Shows significant effort and interest of participants
Results: open benchmarks

- Can compare results of three tools on the same code:
  - tools: aIT, Bound-T, OTAWA
  - benchmarks: debie1, PapaBench
  - processor: ARM7 with single-cycle memory access

- Some differences are seen, reasons not always clear...
  - differences in flow analysis? eg. infeasible paths
  - different interpretations of Analysis Problem conditions?
  - differences in processor modelling?

- Comparison to debie1 measured ET from Tidorum
  - measurements came very late (not originally planned)
  - **NOT** worst-case measurements, in general
  - still, comparison to aIT & Bound-T results revealed errors:
    - in analysis annotations (aIT and Bound-T)
    - in measurement procedures (debie1 harness code)
Example of results for ARM7

Note: “measured” is generally **not** the worst case!

<table>
<thead>
<tr>
<th>BM problem</th>
<th>aiT</th>
<th>Bound-T</th>
<th>OTAWA</th>
<th>measured</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>debie1:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-T1</td>
<td>342</td>
<td>333</td>
<td>332</td>
<td>303</td>
</tr>
<tr>
<td>2a-T1</td>
<td>100</td>
<td>93</td>
<td>139</td>
<td>93</td>
</tr>
<tr>
<td>3a-T1</td>
<td>2 664</td>
<td>2 692</td>
<td>4 101</td>
<td>2 340</td>
</tr>
<tr>
<td>4b-T1</td>
<td>215</td>
<td>214</td>
<td>210</td>
<td>206</td>
</tr>
<tr>
<td>5b-T1</td>
<td>38 798</td>
<td>39 825</td>
<td>55 883</td>
<td>37 463</td>
</tr>
<tr>
<td>6c-T1</td>
<td>40 143</td>
<td>42 285</td>
<td>-</td>
<td>19 726</td>
</tr>
<tr>
<td><strong>PapaBench:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>1 716</td>
<td>1 660</td>
<td>1 358</td>
<td></td>
</tr>
<tr>
<td>A2b</td>
<td>31 482</td>
<td>37 181</td>
<td>38 112</td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>12 051</td>
<td>17 378</td>
<td>17 422</td>
<td></td>
</tr>
</tbody>
</table>
The Daimler Experiment

- Analysts: Stuttgart students (Krause, Geppert, Fellger)
  - no earlier experience of WCET analysis or this benchmark
  - access to Daimler experts on the benchmark code
  - remote support from tool providers

- Tools: aiT, OTAWA (oRange not used)
  - TuBound installation problems at Daimler
  - SWEET feared same, did not try

- Target: MPC5553 (PowerPC; no external memory)

- Benchmark: one program, part of truck control system

- Analysis Problems (all WCET questions):
  - interrupt handler (INTERR), no loops, one call
  - initialization routine (INIT), no loops, no calls
  - calculation routines (CALC1, 2), some loops and calls
  - complete task (TASK), endless loop: call subtasks, suspend
Results: Daimler Experiment

- Students report on **usability**
  - aiT: pretty straightforward to use; no major problems; loop-bound over-estimation reasonable
  - OTAWA: could not handle infinite loops; sometimes crashed; needs oRange for loop-bounds analysis (not used here)

- **Flow analysis**
  - context-dependent loop-bounds important
  - available in aiT, not in OTAWA

- **WCET comparison aiT - OTAWA**
  - actual ETs not measured
  - the target is MPC555\textsuperscript{3}, but OTAWA supports MPC555\textsuperscript{4}
    - aiT analysis was executed for both models
    - OTAWA result usually $<$ aiT result, but...
  - OTAWA results (for other code) sometimes $<$ measured ET
    - OTAWA's model of MPC554 needs correction?
### Daimler Experiment: WCETs

<table>
<thead>
<tr>
<th>Entry point</th>
<th>aiT MPC5553</th>
<th>aiT MPC5554</th>
<th>OTAWA MPC5554</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERR</td>
<td>524</td>
<td>204</td>
<td>113</td>
</tr>
<tr>
<td>INIT</td>
<td>1,055</td>
<td>494</td>
<td>218</td>
</tr>
<tr>
<td>CALC1</td>
<td>2,124</td>
<td>830</td>
<td>722</td>
</tr>
<tr>
<td>CALC2</td>
<td>16,507</td>
<td>6,218</td>
<td>7,991</td>
</tr>
<tr>
<td>TASK</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WCET for “TASK” not reported because this entry point has an infinite loop.
Summary of WCC'11

**Successes** (goals set after WCC'08):
- more participants, in particular many from WCC'06 again
- some comparable results:
  - same processor, same benchmark, different tools
- some continuity in benchmarks and targets
- but also new participants, targets, benchmarks
- evaluation of usability with novice users (Daimler experim.)

**Failures**:
- not many comparable results
- no validated WCETs for reference
  - measured ETs for only one benchmark
- few participants for the “complex processor” (MPC)
- few participants in the Daimler experiment
  - non-disclosure requirements, hinder eg. porting
- Worst Case Challenge is still sporadic, not continuous
Reactions from WCC'11 participants

- “Our research benefits from the extended pool of benchmarks” (FORTAS)
- “[now] much clearer [how] to improve the tool” (METAMOC)
- “a good source of inspiration” (TimeWeaver)
- “we got many new ideas for improving the tool” (WCA)
Suggestions for future Challenges

- Split into two phases?
  1. Flow analysis phase, giving loop bounds etc.
     - combine to give best (tightest) flow constraints
     - eg. only SWEET could find bounds on floating-point loops
  2. WCET analysis phase
     - all tools use the same (best) flow information
     - separate flow analysis from processor modelling

- Several participants supported the two-phase idea

- Usability evaluation ("Daimler Experiment")
  - should be at start of Challenge
    - to give tool providers time to respond
  - should preferably use open benchmark
    - so that experiences can be discussed openly

- Better support for measurement-based tools: test cases!
Thanks

Thanks to all Challenge participants!