

# **VM Rumble: CACAO**

Christian Thalinger  
Vienna University of Technology  
`twisti@complang.tuwien.ac.at`

## What we did last year... (1)

- OpenJDK as Java core library support
  - separate IcedTea branch
    - \$ hg up -C cacao
    - \$ ./configure --with-cacao
  - run OpenJDK on any architecture CACAO supports
- JIT ports
  - PowerPC64 and m68k (Roland Lezuo)
  - s390 (Peter Molnar)
- exact GC, not finished yet (Michael Starzinger)
  - GC/replacement points
- LLNI (low-level native interface) changes (Michael Starzinger)
  - to support moving GC

## What we did last year... (2)

- annotation support (Mathias Panzenboeck)
- assertion support (Gregor Kaufmann)
- internal JIT compiler changes
  - exception table, line-number table
  - using hardware-traps for
    - \* implicit exceptions
    - \* code patching
    - \* JIT compiler calls
- threading code rewritten
- stack-trace code rewritten
- started a system function call abstraction layer to be more portable

## What we are currently doing...

- JIT ports
  - SPARC64 (Alexander Jordan)
- exact GC (Michael Starzinger)
- integrating (again) SSA and LSRA (Peter Molnar)
  - rewriting
  - working then on escape analysis
- replace internal UTF8 strings with `java.lang.String` compatible ones (Andreas Hubert)
- exception handling in C code

## What we plan to do...

- lock inlining
  - already have some code
  - portable way to get TLS (using two different stacks)
- biased locking
- implicit-exception check removal (Ph.D.)
  - instruction scheduling and code motion
- backend generation (Ph.D.)
  - two different backends
    - \* a fast one for the baseline compiler (very similar to the current one)
    - \* a optimizing one (working with LIR and MIR)
- garbage collecting classes and JIT-code