

VM Rumble: CACAO

Michael Starzinger

Agenda

- What we did last year
- What we are currently doing
- What we plan to do
- The future of CACAO

The past year ... (I)

- Added abstraction for signals
 - too many 'mcontext' structure layouts
 - use machine-independent 'executionstate'
- Switched to C++ (Christian Thalinger)
 - type-safe object accessors (replaces LLNI macros)
- Floating-Point Acceleration (Michael Starzinger)
 - support for Vector Floating-Point (VFP) on ARM

The past year ... (2)

- **Escape Analysis** (Peter Molnar)
 - researched impact of thread local heaps
- **Fast Subtype Checking** (Stefan Ring)
 - no class renumbering
 - it actually works on multi-core machines
- **Global References unified** (Michael Starzinger)
 - using global references internally now
 - one implementation for internal and JNI usage

The past year ... (3)

- JVMTester implemented (Christian Thalinger)
 - nightly builds of VMs with GNU Classpath, OpenJDK & PhoneME
 - provides test results of Mauve, JTReg, SpecJVM & Dacapo Benchmarks

Work in progress (I)

- Lock Inlining

- split into fast-path and slow-path

- (Re)designing the Stack Layout

- safely throw a `StackOverflowException`
- fast access to TLS (or 'threadobject' in our case)
- get rid of reentrant signals

Work in progress (2)

- Exact Garbage Collector
 - switching to handles (done!)
 - safe points (currently only replacement points)
- JIT-Cache for ARM port
- Merge String and UTF8 handling code

Roadmap

- Collection of JIT Code
- Further GC improvements
 - support for Weak-, Soft- & Phantom-References
 - add support for generational collection
 - class unloading
- Low-level IR
 - aims at backend generation

The future of CACAO

- What CACAO is focusing on a.t.m.
 - fast JIT for embedded systems (ARM, PPC, MIPS, ...)
 - research VM with 'readable' code
- What should be the focus for the future?
 - running on barebone microkernel (L4, Hurd, ...) without Linux?
 - remains an open question!
- We will try to pass the JCK!

Thank You!