Jitrino OPT

List of IR transformations used in Jitrino.OPT compiler:

Use this link for details on type system used in Jitrino.OPT compiler

IR construction:

- translator - Translates Java bytecode to High Level Intermediate Representation (HIR)
- hir2lir - Translates HIR to Low Level Intermediate Representation (LIR)

High-Level (HIR) optimizaitons

- optimizer - An alias for all HIR transformations
- abce - Array bounds check elimination
- dce - Dead code elimination
- dessa - SSA deconstruction
- devirt_intf - Guarded devirtualization of interface calls
- devirt_virtual - Guarded devirtualization of virtual calls
- edge_annotate - IR annotation with edge profile
- edge_instrument - IR instrumentation in order to collect edge profile
- escape - Escape analysis bases optimizations
- gcm - Global code motion
- gvn - Global value numbering
- hlo_api_magic - Replaces known Java API method calls with predefined algorithms written with HIR
- hvn - Hash value numbering (CSE)
- inline - Inlines hot methods calls
- inline_helpers - Replaces HIR instruction with a Java method call and inlines this call
- lazyexc - Eliminates redundant creation of exception objects
- lower - Partial inlining for type checks
- memopt - Redundant memory loads and stores elimination
- osr - Operator strength reduction
- peel - Loop peeling
- purge - Purge empty nodes
- simplify - Type/copy propogation and constant folding
- ssa - SSA construction
- statprof - Annotates HIR with edge profile using static heuristics
- throwopt - Replaces Throw instruction with the jump to the exception handler
- uce - Unreachable code elimination
- unguard - Removal of untaken type guards
- unroll - Loop unrolling

Low-level (LIR) optimizations

- codegen - An alias for all LIR transformations
- api_magic - Replaces known Java API method calls with predefined algorithms written with LIR
- bbp - Back branch polling for loops
- break - Inserts 'int3' into the prologue of the generated method
- btr - Branch optimizations
- cafl - Complex address form loader
- cg_dce - Removes dead code
- cg_fastArrayFill - Finds and replaces particular internal helper with a loop providing fast array filling with a constant
- constraints - Performs resolution of operand constraints and assigns calculated constraints to operands
- copy - translates CopyPseudoInsts to corresponding copying sequences
- early_prop - A simple algorithm of constant and copy propagation
- emitter - Emits binary code from LIR
- gcmap - Builds a map with information for all Object and Managed pointers operands locations on suspension points
- gcpoints - Inserts pseudo-use instructions for some Object operands to extend their live-ranges.
- i586 - Replaces SSE2 instructions with SSE or FPU ones if SSE2 is not supported by the current platform
- i8l - Splits 64-bit integer operands into two 32-bit ones
- info - Registers various method information in VM memory associated with the method
- iprof - Instruments a method with various profile collection counters
- itrace - Instruments method entry/exit and unwind points with calls to logger
- layout - Prepares code layout
- light_jni - Allows to call some predefined JNI methods directly avoiding JNI stub
- native - Transforms 3-address LIR form into 2-address form
- peephole - Performs various architecture specific per-inst optimizations
- rce - Removes redundant comparisons. The analysis is based on EFLAGS affect
- regalloc - Global register allocator
- si_insts - Saves information about call instructions for stack unwinding
- spillgen - Local (basic-block level) register allocation and spill generator
- stack - Layouts stack and assigns locations for operands on stack

Auxilary runtime and confuguration services:

- opt_init - Initializes global optimizer flags
- lock_method - Locks method data in VM
- unlock_method - Unlocks method data in VM