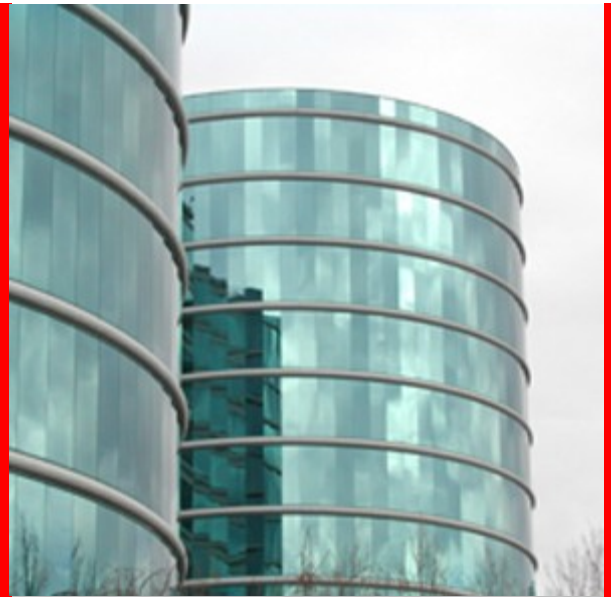

ORACLE®

Tiered Compilation

Igor Veresov

Goals

- Server performance
- With client startup
- Adaptive policy with minimal or no machine-dependent tuning: feedback on compilation queue length as a measure of machine's speed/compilation speed

Compilation levels

- level 0 - interpreter
- level 1 - C1 with full optimization (no profiling)
- level 2 - C1 with invocation and backedge counters
- level 3 - C1 with full profiling (level 2 + MDO)
- level 4 - C2

- level 3 doesn't do CEE or BE
- level 3 is 35% slower than level 2

Client is hard to beat

- Client compiles when i + b >= 1500 (level 1), needs no profiling
- Tiered needs to compile in best case:
  - level 3 version and profile (code 35% slower)
  - compile level 4 version (compilation is slow)
- Only beneficial for workloads when it's possible to compensate by producing level 4 version fast
- Fortunately most real word apps (and benchmarks) are like that (run more than a second with high code reuse factor).

Profiling

- Profiles stored in MethodDataOop (MDO)
- Existing counters used when not profiling(MethodOop)
- Separate counters in MDO (multiple versions can run concurrently)
- Can profile at levels 0 and 3
- What: branches, call receiver types, typechecks (checkcast, instanceof, aastore)
- Periodic runtime notifications.
  - Invocations: 0:128, 2:2048, 3:1024,
  - Backedges: 0:1024, 2:8192, 3:4096

State transitions I

- Starts at level 0
- Ideally needs to transition to level 3
- Scales threshold based on C1 queue length / number of C1 threads
- May transition to level 2 based on C2 queue length (spending too much time in level 3 hurts, 30% slower)
- Can start profiling at level 0

State Transitions II

- Profiling: level 3
- Monitors C2 queue length / number of C2 threads and scales level 4 threshold
- If the method is trivial (small, nothing to profile) compiles level 1 version immediately

In-queue optimizations

- Prioritization by (rate + 1) * (i + 1) * (b + 1),
  rate = d(i + b) / dt
- And methods after deoptimization have advantage
- Stale methods removal
- In-queue level change (3->2).

Common transition patterns

- 0 -> 3 -> 4 (common).
- 0 -> 2 -> 3 -> 4 (C2 queue too long).
- 0 -> (3->2) -> 4 (in queue change).
- 0 -> 3 -> 1 or 0 -> 2 -> 1 (trivial, or can't compile in C2).
- 0 -> 4 (can't compile in C1, fully profile in interpreter).
- (1,2,3,4) → 0 (deoptimization)

Level switching predicates

- Regular compile:
  i > k1 * scale || i > k2 * scale && i + b > k3 * scale
- OSR: b > k4 * scale
- Profile maturity measured the same way
  (scale=ProfileMaturityPercentage/100.0 = 0.2)
- Reprofiling support (additional start counters in MDO)

Performance (8 core x86, 1 C1, 2 C2)

Client vs tiered

benchmark: _227_mtrt
startup: 1.857 1.692 diff: 8.88%
settled: 0.5961 0.4107 diff: 31.10%
benchmark: _202_jess
startup: 4.066 3.615 diff: 11.09%
settled: 1.4405 1.2545 diff: 12.91%
benchmark: _201_compress
startup: 4.781 4.467 diff: 6.56%
settled: 3.9605 3.6429 diff: 8.01%
benchmark: _209_db
startup: 9.719 8.863 diff: 8.80%
settled: 6.2362 6.3588 diff: -1.96%
benchmark: _222_mpegaudio
startup: 2.656 2.424 diff: 8.73%
settled: 2.4167 2.2408 diff: 7.27%
benchmark: _228_jack
startup: 3.737 3.339 diff: 10.65%
settled: 1.7389 1.2857 diff: 26.06%
benchmark: _213_javac
startup: 3.714 3.649 diff: 1.75%
settled: 2.0299 1.6375 diff: 19.33%

Server vs tiered

benchmark: _227_mtrt
startup: 2.014 1.692 diff: 15.98%
settled: 0.4234 0.4107 diff: 2.99%
benchmark: _202_jess
startup: 4.633 3.615 diff: 21.97%
settled: 1.3518 1.2545 diff: 7.19%
benchmark: _201_compress
startup: 4.942 4.467 diff: 9.61%
settled: 4.1837 3.6429 diff: 12.92%
benchmark: _209_db
startup: 9.003 8.863 diff: 1.55%
settled: 6.4088 6.3588 diff: 0.78%
benchmark: _222_mpegaudio
startup: 3.035 2.424 diff: 20.13%
settled: 2.2615 2.2408 diff: 0.91%
benchmark: _228_jack
startup: 4.911 3.339 diff: 32.00%
settled: 1.5041 1.2857 diff: 14.52%
benchmark: _213_javac
startup: 5.681 3.649 diff: 35.76%
settled: 1.7092 1.6375 diff: 4.19%

SPECjvm98, each sub-benchmark ran in a fresh JVM for 20 iterations.
startup – first iteration, *settled* – average of last 10.