Performance Anxiety

Joshua Bloch
Google Inc.

Disclaimer

> This is a talk about performance
  - Premature optimization is evil
  - Don’t do it

> But occasionally you do have to optimize
  - Especially if you build libraries, languages, or VMs

The Basic Problem

- Once upon a time, you could estimate performance by counting instructions
- Today it is *impossible* to estimate performance
  - You have to measure it
  - Programming has become an empirical science
- The cause: increased abstraction gap between programs and their execution
- True at every level
  - Library, language, VM, processor

And It Gets Worse...

Demo

Yep, You Saw That Right

> Times are consistent from run to run in a single “program-run” (after the VM is warmed up)
> But restart the VM and you get a different result!
> Makes it terribly difficult to measure the effects of performance changes

What's Going On?

- **Absolute performance and performance model quality are inversely related**
  - Because complexity, predictability are inversely related
- After you’ve picked the low hanging fruit, more performance comes at the cost of more complexity

But What’s *Really* Going On?

- No one knows
- The software/hardware stack is too complex
- People used to blame it all on cache misses
- But Cliff Click notes that *compilation planning* often to blame

What's Compilation Planning?

> Suppose this is a hot loop; what does VM inline?

```
static void foo(int m, int n) {
    for (int i = 0; i < m; i++)
        bar(n);
}

static void bar(int n)  {
    for (int i = 0; i < n; i++)
        baz();
}
```

Compilation Planning (Continued)

- CP thread uses heuristics to decide what to inline
  - Decisions based on local info, have global impact
- In previous example, inlining outer loop may preclude inlining inner loop and vice-versa
  - We know it's best to inline inner loop; VM may not
- Because decision is made in a background thread, it's non-deterministic – could vary from run to run
- This might explain observed variation
  - But who knows?

Surprising Behavior

- Which is faster, conditional AND (`&&`) or logical (`&`)?
- In the old days, conditional or was faster
  - Avoided evaluating the 2nd operand if unneeded
- Now logical AND is often faster, as it is *branch-free*
  - Modern processors do *branch prediction*
  - Mispredicted branches are expensive (~100 cycles)

Unpredictable Behavior

- What's the cost of field access (`obj.field`)?
  - It depends
- Is field volatile? If so, what processor are you using?
  - ARM, Power are problematic
- Is field private on a nested class?
  - You're doing a method invocation!

Are There More Examples Like This?

> Is the Pope Catholic?
  - Previous examples barely scratched the surface

> Problem isn't going away; it's getting worse

> It’s a matter of “software physics”
  - Predictability α 1/Complexity
  - Corrolary: Performance-model α 1/Performance

What Can You Do?

- **If you're an application programmer**
  - Use high-level, declarative constructs where possible
    - Let lower levels make things as fast as possible
  - Learn to live with the unpredictability
- **If you're a library, language, or VM programmer**
  - Learn to avoid “sources of astonishment”
    - e.g., make nested classes package-private, not private
  - **Measure, measure, and measure again**
  - Reimplement when underlying perf model changes

What Others Are Saying About This

> **In Java, run VM 40 times to get meaningful data**

- Georges, Buytaert and Eeckhout, *Statistically Rigorous Java Performance Evaluation*. OOPSLA '07

> **The problem doesn't go away if you use C or C++**

- Mytkowicz and Diwan, *Producing Wrong Data Without Doing Anything Obviously Wrong!*. ASPLOS '09

> **It doesn't even go away if you use assembly code**

- Cliff Click and Brian Goetz, *This Is Not Your Father's Von Neumann Machine*. JavaOne ‘09

> **We aren't teaching this stuff, but we should be**

- Doug Lea, David Bacon, and David Grove, Languages and Performance Engineering: Method, Instrumentation, and Pedagogy (2008 SIGPLAN Workshop on PLC)

Conclusion

> We live in interesting times
> Performance anxiety is a fact of life
> But we can (and must) learn to live with it

Performance Anxiety

Joshua Bloch
Google Inc.