Language Performance

4-16-2010

Opening Discussion

- Do you have any questions about the quiz?
- What did we talk about last class?

Compilers, Interpreters, and JITs

- Machines don't understand the code you write. Something most convert it to something the machine understands.
  - Compilers take your code to machine language (often with a step are assembly in between).
  - Interpreters are programs that parse text and execute it on the fly. Interpreters are slow.
  - JIT (Just In Time) Compilers take text or bytecode and make machine language at runtime.

Optimization

- Because of the complexity of modern architectures, the performance of machine code can be greatly impacted by optimizations used by the compilers.
- Poor optimization can produce slow code from any language.
  - In C/C++ make sure you compile with optimization turned on.
- Poor programming practice can give you code that is even slower.
  - For example passing structs/classes in C/C++ by value instead of by reference.

Comparing Some Languages

- Matlab and Perl are both interpreted languages with JITs.
- Other languages you might know.
  - C/C++ - Compiled languages that expose the underlying machine. Let's the coder optimize, but makes life harder for the compiler.
  - Java/C#/... - Compiled to bytecode

Benchmarks

- http://shootout.alioth.debian.org/
- This site maintains some nice benchmarks using different languages and compares performance.
- They go to lengths to point out this approach is a bit flawed, but it is still quite interesting and we can learn some things from it.

Closing Remarks

- Have a good weekend.
- Play some with assignment #8. If it sucks too much send me an e-mail so I can come up with something else.