ECE 571 – Advanced Microprocessor-Based Design Lecture 2

Vince Weaver

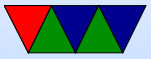
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

2 September 2020

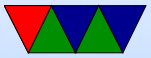
Announcements

- Lecture notes are posted to the course website.



Review: What is Performance?

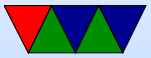
- Getting results as quickly as possible?
- Getting *correct* results as quickly as possible?
- What about Budget?
- What about Development Time?
- What about Hardware Usage?
- What about Power Consumption?
- What about Security?



Motivation for HPC Optimization

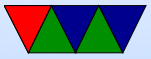
HPC environments are expensive:

- Procurement costs: $\sim\$40$ million
- Operational costs: $\sim\$5$ million/year
- Electricity costs: 1 MW / year $\sim\$1$ million
- Air Conditioning costs: ??

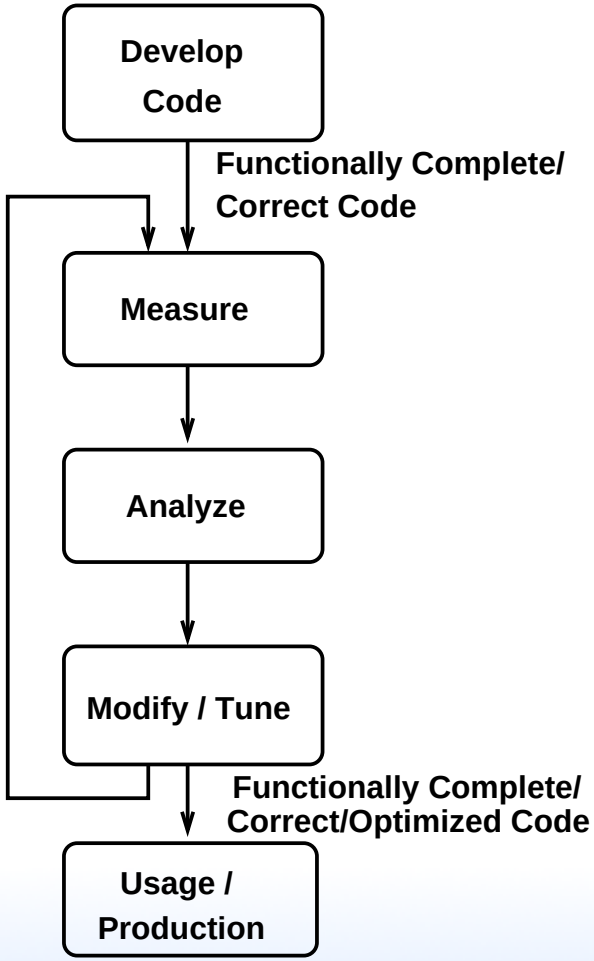


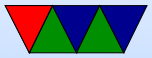
Know Your Limitation

- CPU Constrained
- Memory Constrained (Memory Wall)
- I/O Constrained
- Thermal Constrained
- Energy Constrained



Performance Optimization Cycle



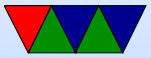


Wisdom from Knuth

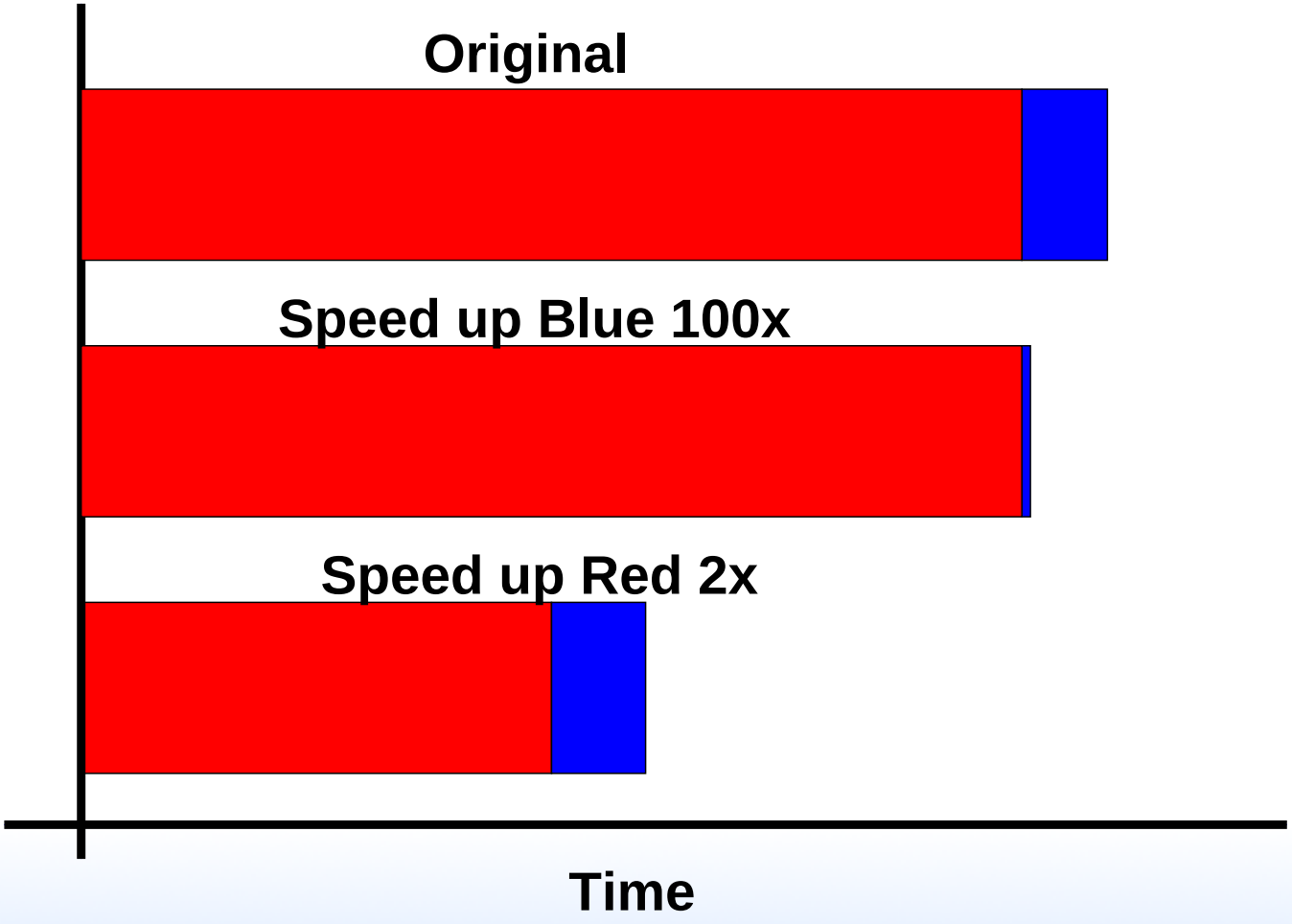
"We should forget about small efficiencies, say about 97% of the time:

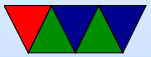
premature optimization is the root of all evil.

Yet we should not pass up our opportunities in that critical 3%. A good programmer will not be lulled into complacency by such reasoning, he will be wise to look carefully at the critical code; but only after that code has been identified" — Donald Knuth

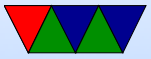


Amdahl's Law



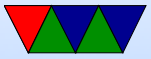


Software Tools for Performance Analysis



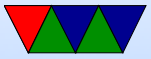
Simulators

- Architectural Simulators
- Can generate traces, profiles, or modeled metrics
- Slow, often 1000x or more slower
- Not real hardware, only a model
- Did I mention, slow?
- m5, gem5, simplescalar, etc



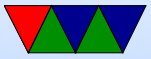
Dynamic Binary Instrumentation

- Pin, Valgrind (cachegrind), Qemu
- Still slow (10-100x slower)
- Can model things like cache behavior (can model parameters other than system running on)
- Complicated fine-tuned instrumentation can be created
- Architecture availability – Pin (no longer ARM), Valgrind, Qemu most architectures, hardest to use



Compiler Profiling

- gprof
- gcc -pg
- Adds code to each function to track time spent in each function.
- Run program, gmon.out created. Run "gprof executable" on it.
- Adds overhead, not necessarily fine-tuned, only does time based measurements.
- Pro: available wherever gcc is.

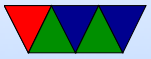


Gathering Performance Info – Aggregate Counts

- Aggregate counts (total instructions, total cycles, etc)
- Actual measurements: `perf`, `time`

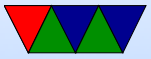
- DBI measurements: `valgrind`, `qemu`

- Simulators: `gem5`, `simplescalar`



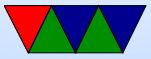
Gathering Performance Info – Profiling

- Insert calls on entry to function (or basic block) to track how much time spent in each
- Do you need source code?
- Manually add?
- DBI: `valgrind`
- compiler: `gprof`



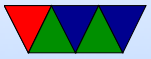
Gathering Performance Info – Sampled Counts

- Sampled counts – periodically interrupt program, note the instruction pointer
- Can use info to statistically determine which part of code where most time (or other metric) is spent
- hardware: `perf`
- DBI: `valgrind`



Gathering Performance Info – Tracing

- Tracing – gather a record of every event (instruction?) that is executed. Can then replay this trace through various tools for analysis.
- Downside: huge trace files (gigabytes+)



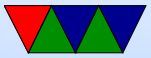
Performance Data Analysis

Manual Analysis

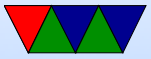
- Visualization, Interactive Exploration, Statistical Analysis
- Examples: TAU, Vampir

Automatic Analysis

- Try to cope with huge amounts of data by automatic analysis
- Examples: Paradyn, KOJAK, Scalasca, Perf-expert

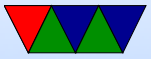


Evaluating Performance of Modern Systems



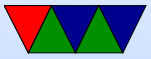
Benchmarks

- When measuring performance, need a reference workload to compare
- Ideally reproducible, portable, easy to compile, relevant
- Benchmarks can be gamed



Selected Commonly Seen Benchmarks

- SPEC
  - CPU 2000, CPU 2006, CPU 2017 – Commercial, Single-threaded (floating point and integer)
  - OMP – Commercial, Parallel
  - jbb – Java
- HPC Challenge – Free. HPL (Linpack). High-performance / Linear Algebra
- HPCG (conjugate gradient) new replacement for HPL
- PARSEC – Free, Multithreaded / CMP



- MiBench – Free, Embedded (2000)
- BioBench, BioParallel – Free, Bio/Data-Mining
- lmbench – Free, Operating System
- iobench – Disk I/O
- Stream – Memory