Project
Algorithm Engineering

Jens K. Mueller

jkm@informatik.uni-jena.de

Department of Mathematics and Computer Science
Friedrich-Schiller-University Jena

Wednesday 7th May, 2014

Machine-dependent Optimizations

Least Squares

- Given m observations $\{(x_1, y_1), \cdots, (x_m, y_m)\}$ the (linear) least squares problem seeks to find the minimizer to

$$f(s, b) = \sum_{i=1}^{m} (s \cdot x_i + b - y_i)^2$$

- Closed form solution

$$s = \frac{\sum_{i=1}^{m} x_i y_i - \frac{1}{m} \sum_{i=1}^{m} y_i \sum_{i=1}^{m} x_i}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m} (\sum_{i=1}^{m} x_i)^2}$$

$$b = \frac{1}{m} \sum_{i=1}^{m} y_i - s \cdot \frac{1}{m} \sum_{i=1}^{m} x_i$$

Latency and Issue Time

- Latency bound
  Lower bound on cycle/element for data-dependent instructions in sequence
- Throughput bound
  Lower bound on the number of functional units

Latency and Issue Time

An instruction has

- Latency time (typically in cycle)
  Time until results available
- Issue time (typically in cycle)
  Number of cycles (between two successive) instructions

- Latency bound
  Lower bound on cycle/element for data-dependent instructions in sequence
- Throughput bound
  Lower bound on the number of functional units

CPU-specific Optimization Information

- Vary for different CPUs
- But similar for CPU generations

- CPU vendors
  Intel 64 and IA-32 Architectures Optimization Reference Manual
- Researchers
  Agner's Optimization manuals

Manual Loop Unrolling

- Reduce loop overhead
  Operations that do not contribute

- Don't miss the last iterations
  You need to fix up the last unrolled iterations.

Critical Path

- Data-dependent instructions in loops
- Running time is dominated by latency of instruction
- Reduce data dependencies by introducing temporary variables (registers)

Instruction Level Parallelism

- Breaking data-dependencies allows executing instructions in parallel or at least pipelined
- Optimizes throughput

Reassociation

- Compiler is free use to associativity for integer addition and multiplication
- Associativity does not hold for floating point numbers

- Compiler switch `-fassociative-math`
- Do it manually

Register Spilling

- Introducing additional registers to break dependencies
- No registers available, then put on stack (expensive)

Branch Prediction

- Measure branch prediction
  Usually CPUs are pretty good at
- But branches may be random
- Avoid branching or use conditional moves

Homework

- Compute CPE [cycle/element]
- Measure for your fastest solution
- Apply loop unrolling
- Enhance instruction level parallelism (critical path)