Algorithm Engineering

Jens K. Mueller

jkm@informatik.uni-jena.de

Department of Mathematics and Computer Science
Friedrich Schiller University Jena

Monday 19th January, 2015

Machine-dependent Optimizations

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 fully 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