Merrimac Architecture Comparison Methodology (Discussion)

Mattan Erez

Merrimac meeting

January 20, 2004

Comparison Methodology

- Metrics
- Systems to compare to
- Benchmarks
- Suggested methodology
- Questions
  - The dark side of benchmarking
  - Simulation framework

What are Ops?

- All operations executed
  - Floating Point
  - Integer
  - Loads/Stores
    - This will not lead to fair comparisons
- Just essential operations
  - As defined by the algorithm
  - As defined by the implementation
- What about divides
  - Trig functions, exponents, …

"Architecture independent" metrics

- Application properties
  - Parallelism
    - max ops/cycle
    - Parallel/Serial ratio
  - Actual parallelism
    - Actual measurements
    - Performance vs. number of nodes
  - Max arithmetic intensity (ops/theoretical I/O word)
  - Actual arithmetic intensity (ops/memory-word)
- System properties
  - Max parallelism
  - Max memory BW

    Use micro-benchmarks

Performance metrics

- Performance
  - Ops/cycle
  - Ops/second
  - Global updates/cycle(second)
- Normalize to
  - Cost
    - Not real cost, but estimates like we have for Merrimac
  - Peak memory BW
  - Peak FLOPs
    - Need to account for different technologies and design effort
  - Power
- Mix of the above
  - I.e. Ops/s/Watt

What systems should we compare to?

- Single node
  - Pentium4/Opteron/PPC/Cray
  - How to calibrate technology?
    - If we wait a few months Intel will be at 90nm
  - What memory system
    - Our memory system is based on faster parts
- Multi node
  - Clusters/SMPs/Clusters-of-SMPs
  - Real system or SVM?
    - Coherency issues
    - Network simulation
    - Simulation run times
    - OS and I/O

Benchmarks – the real problem

- Standard benchmarks
  - Standard supercomputing benchmarks
    - What are they
- Applications that have been optimized on other systems
  - FFT, LINPACK, GROMACS, FLO, sparse algebra
  - Need to port to Merrimac and optimize well
- Our applications
  - Need to port to other system
  - What will be a fair optimization effort?
    - Our compiler doesn't do any optimizations yet
    - Our programming team knows the system better than the comparison system

Benchmarks (2)

- Micro benchmarks
  - Test system capabilities
  - Compute intensive
  - Memory (I/O) intensive
  - Multi-node
    - Expose sharing problems
    - Load balancing
- Do these expose the hard problems/are they convincing?
- Again, what is a fair application optimization
  - Ideally, have equivalent Brook→Merrimac and Brook→P4 compilers and use a single source
- Still not fair
  - Shouldn't we start with a fixed problem and choose the best algorithm for each system?

Suggested methodology

- Main metrics
  - Essential Ops/BW
  - Essential Ops/second/$ (estimate cost)
- Comparison systems
  - Pentium4 (or Opteron) – hopefully we can find a 90nm P4
    - Use VTune unless we can find a good simulator
  - Pentium4 "cluster" workstation
    - A real SMP system
  - Pentium4 "cluster" supercomputer
    - Using similar simulation methods as SVM simulator
- Benchmarks
  - Highly optimized existing Pentium4 code
  - Specialized micro-benchmarks
  - streamFLO (FLO82), streamMD (GROMACS)
  - streamFEM and streamSPAS
    - Need an expert to optimize on P4

The dark side of benchmarking – parameter tuning and application selection

- Should we perform parameter tuning
  - Based on the benchmarks we are going to run
  - How much time should we spend on it
- What parameters
  - Number of functional units
  - Size of register files, SRF, caches
  - Memory system
    - Memory size, number of AGs, …
- Unfair?
  - Our comparison system was not tuned for these benchmarks or applications
  - Isn't that the point?
  - What about re-examining the algorithms

Is our simulation framework good enough?

- Cycle accurate single node stream unit
- Scalar processor not cycle accurate
  - Cycle accurate memory system
  - Other than that 1 instruction/cycle
- Multi node not cycle accurate
- No OS and I/O simulated
- What frequency are we targeting
  - Should we normalize for process and design effort?
  - Currently
    - 90nm 1GHz Merrimac
    - 130nm 3.2GHz Pentium4

Points brought up during the meeting

- What are we trying to learn
  - Identify bottlenecks
  - Convince ourselves and others that Merrimac is a good idea
  - This is not an evaluation of the "essence of streaming"
- Wall-clock time to solution of problem
  - More important than Ops
  - But Ops are necessary for "conventional" comparison
- Complex ops (divides …) can be compensated for
- Can project other systems architecture to match our timeframe
- Do some dense algebra to make sure we can beat ATLAS or GOTO
- Perform some tuning of the architecture
- Single-node before multi-node