Mitigating Amdahl's Law through EPI Throttling

Murali Annavaram, Ed Grochowski, John Shen, Intel-MRL.

ISCA-2005

Special Acknowledgements to:

Bryan Black, Carole Dulong, Richard Hankins, Konrad Lai, Norman Oded , Ravi Rajwar, Ryan Rakvic, Ronny Ronen, Mike Upton, Hong Wang and Uri Weiser

Three Trends

- CMPs will become ubiquitous
  - All platforms, server to mobile
- Software will become increasingly multithreaded
  - Execution time = Sequential time + Parallel time
  - Speedup limited by sequential phase (Amdahl's law)
- Power is a first-order design constraint
  - Power ~ $Perf^{1.73}$
  - Conflicting power demands for sequential/parallel code
- Minimize execution time of MT programs while keeping power within a fixed budget

EPI – Best of Both Worlds

- For best scalar and throughput performance, vary energy expended per instruction (EPI) based on available parallelism
- P = EPI • IPS
  - P = fixed power budget
  - EPI = energy per instruction
  - IPS = aggregate instructions retired per second
- For a fixed power budget
  - Run sequential phases on high-EPI processor
  - Run parallel phases on multiple low-EPI processors

An Energy Per Instruction Throttle

- Four techniques to vary EPI

Method	EPI Range	Time to Alter EPI	Throttle Action
Voltage/frequency scaling	1:2 to 1:4	100us (ramp Vcc)	Lower voltage and frequency
Asymmetric cores	1:4 to 1:6	10us (migrate 256KB L2 cache)	Migrate threads from large cores to small cores
Variable-size core	1:1 to 1:2	1us (fill 32KB L1 cache)	Reduce capacity of processor resources
Speculation control	1:1 to 1:1.4	10ns (pipeline latency)	Reduce amount of speculation

- Software sees symmetrical multiprocessor
  - Unusual property: individual threads become slower as more threads are run simultaneously, even though net throughput increases!

An EPI Prototype Asymmetric Multiprocessor

- Goal: Demonstrate that a EPI throttled MP outperforms an SMP for the same power budget
- Pentium 4 clock throttle
  - Shut off clock with fixed duty cycle: 12.5%..87.5%
  - Per processor control in an MP
  - Clock throttle does *not* alter actual voltage/frequency!
    - Varying duty cycle has similar performance effect as varying EPI
    - Assume that power is proportional to square of duty cycle
- Thread Affinity
  - Assign a process to a specific CPU

Experimental Configurations

- Base SMP: 4-way 2GHz Xeon, 2MB L3, 4GB Memory, 3 Ultra320 disks
- All four configurations have fixed power
  - Power ≈ CPUs* (duty cycle)^2

CPUs	Effective Frequency	Duty Cycle	Power (normalized)	Performance (normalized)
1P	2GHz	8/8	1.00	1.00
2P	1.5GHz	6/8	1.12	1.06
3P	1.25GHz	5/8	1.17	1.08
4P	1GHz	4/8	1.00	1.00

- 2P/1.5GHz and 3P/1.25GHz run-times adjusted to make power exactly same

AMP Configurations

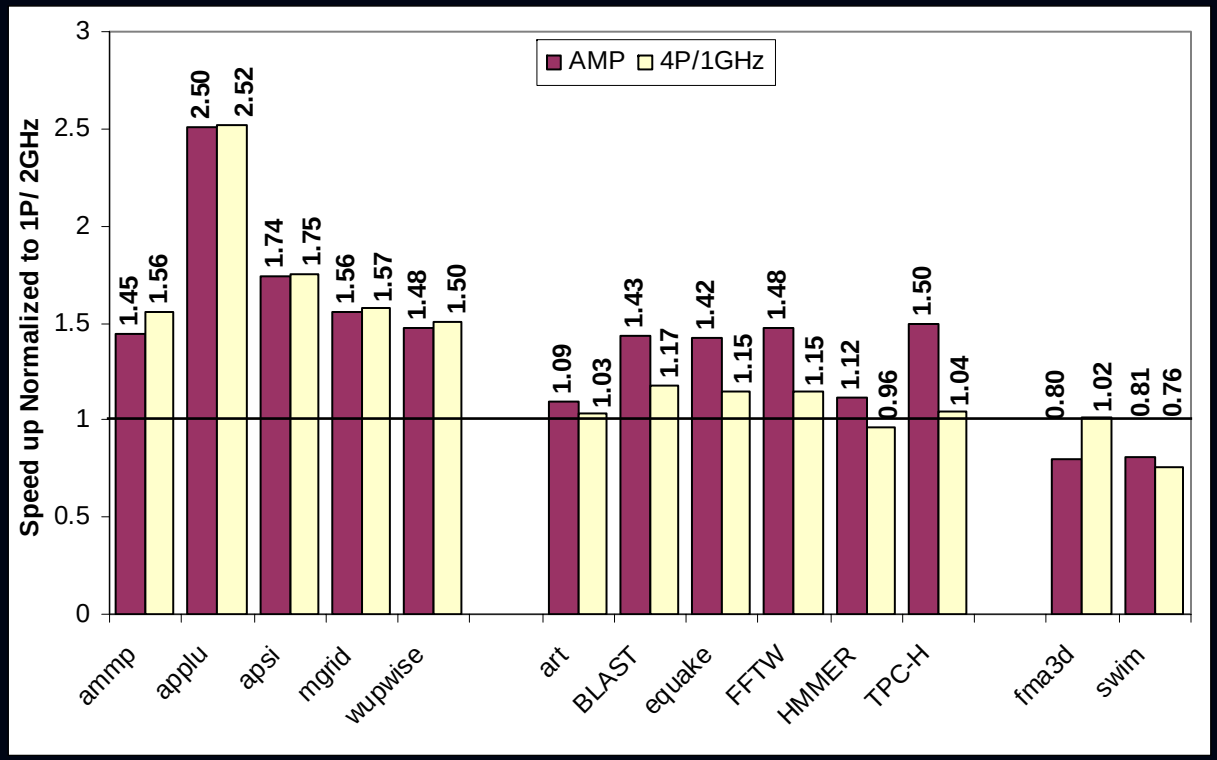
- Static AMP
  - Duty cycles set once prior to program run
  - Parallel phases run on 3P/1.25GHz
  - Sequential phases run on 1P/2GHz
  - Affinity guarantees sequential on 1P and parallel on 3P
  - Benchmarks that rapidly transition between sequential and parallel phases
- Dynamic AMP
  - Duty cycle changes during program run
  - Parallel phases run on all or a subset of four processors
  - Sequential phases of execution on 1P/2GHz
  - Benchmarks with long sequential and parallel phases

Benchmarks

- 13 parallel benchmarks
  - 9 SPEC Open MP benchmarks
  - BLAST & HMMER bio informatics programs
  - TPC-H decision support benchmark
  - FFTW parallel fourier transform solver
- Hand-modified programs
  - OMP threads set to 3 for static AMP
  - Calls to set affinity in each thread for static AMP
  - Calls to change duty cycle and to set affinity in dynamic AMP

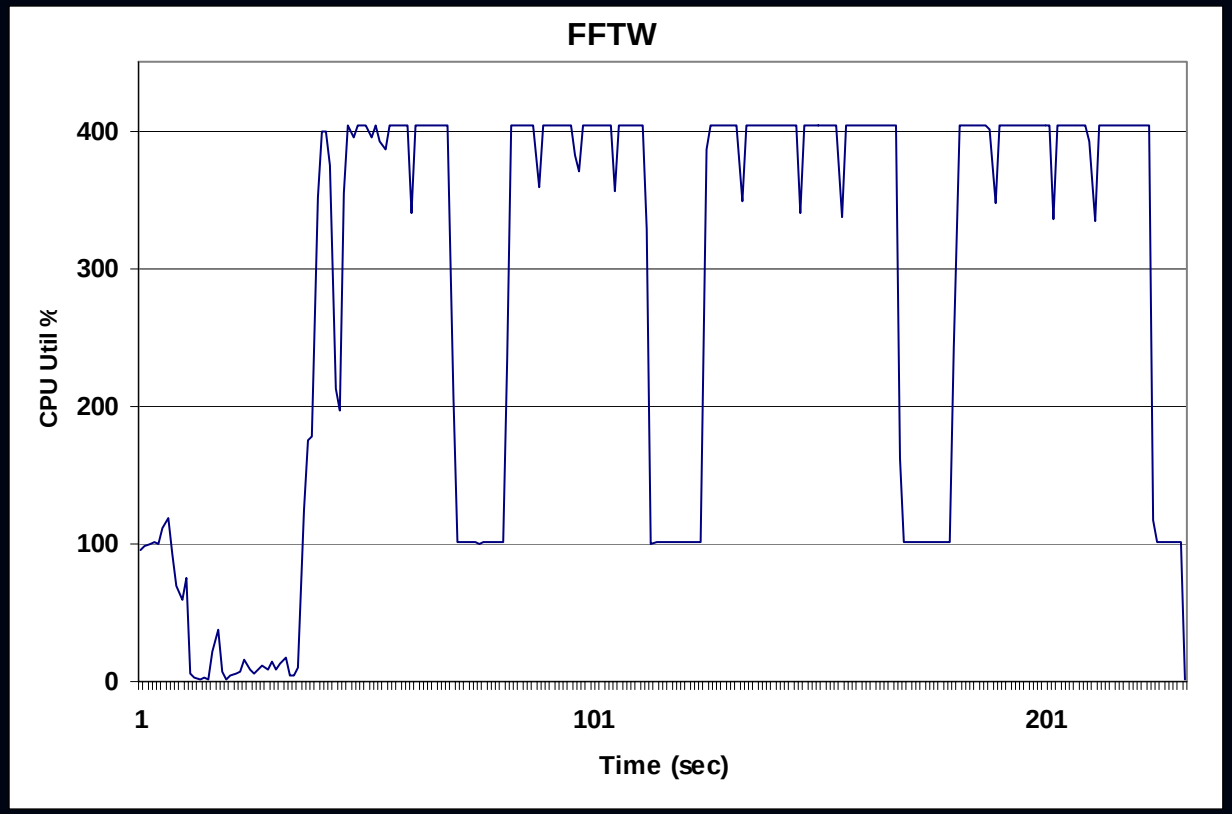
AMP Configuration	Benchmarks
Static AMP	wupwise, swim, mgrid, equake, fma3d, art, ammp, BLAST, HMMER
Dynamic AMP	applu, apsi, FFTW, TPC-H

Speedup on AMP



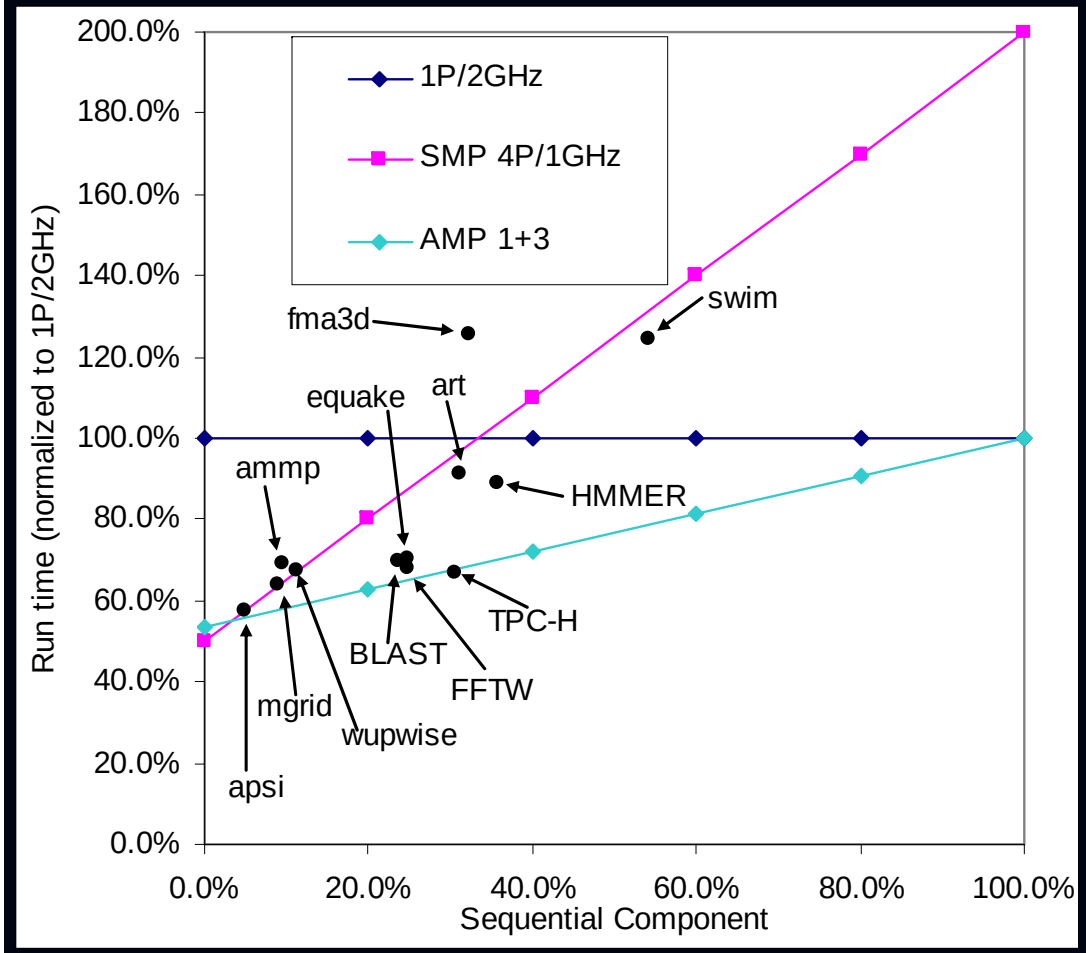
- **Results fall into three categories**
  1. **4P/1GHz SMP and AMP perform equally well**
  2. **AMP achieves significant speedup compared to SMP**
  3. **AMP and 4P/1GHz SMP perform worse**

Intuitive Explanation of Results



- 4P/1GHz underutilizes power during sequential phases
- 1P/2GHz unable to exploit available thread-level parallelism
- AMP varies EPI with TLP to continuously optimize power

Why and When AMP is Better?



- Compute % time in parallel and sequential
- Compare run-times
  - Measured on AMP prototype
  - Projected on ideal AMP
- Clustered in 3 categories
  - Mostly parallel: SMP better
  - Mostly sequential: 1P better
  - Moderately parallel: AMP better

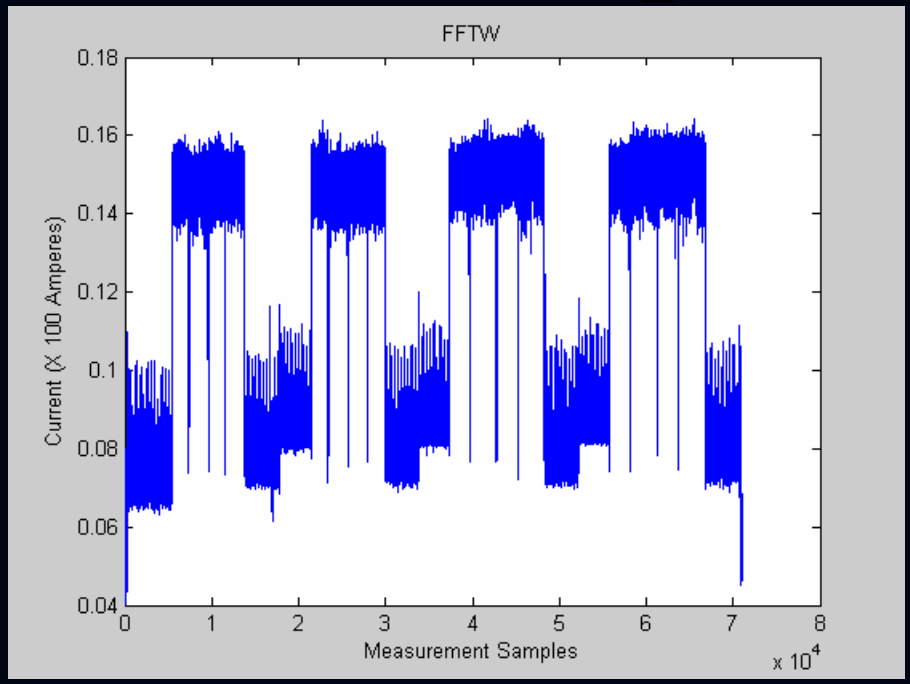
Why AMP Doesn't Always Win

- Benchmark may not have right ratio of parallel/sequential
  - ~100% serial -> Use one fast CPU
  - ~100% parallel -> Use all slow CPUs
- Rapid transitions between parallel and sequential phases
  - Thread migration and throttling overhead
- Benchmark may not be CPU-bound!

An EPI Simulator
Flexible EPI Throttle

- Goal: demonstrate similar results as AMP using completely different method
- Approach
  - Measure current supply on a physical system
  - Use software simulator of EPI throttle
- Why Simulate?
  - CPU power varies continuously
    - AMP monitors CPU power with 1-bit resolution
  - Measure CPU power with 14-bit resolution

Current Measurement Setup

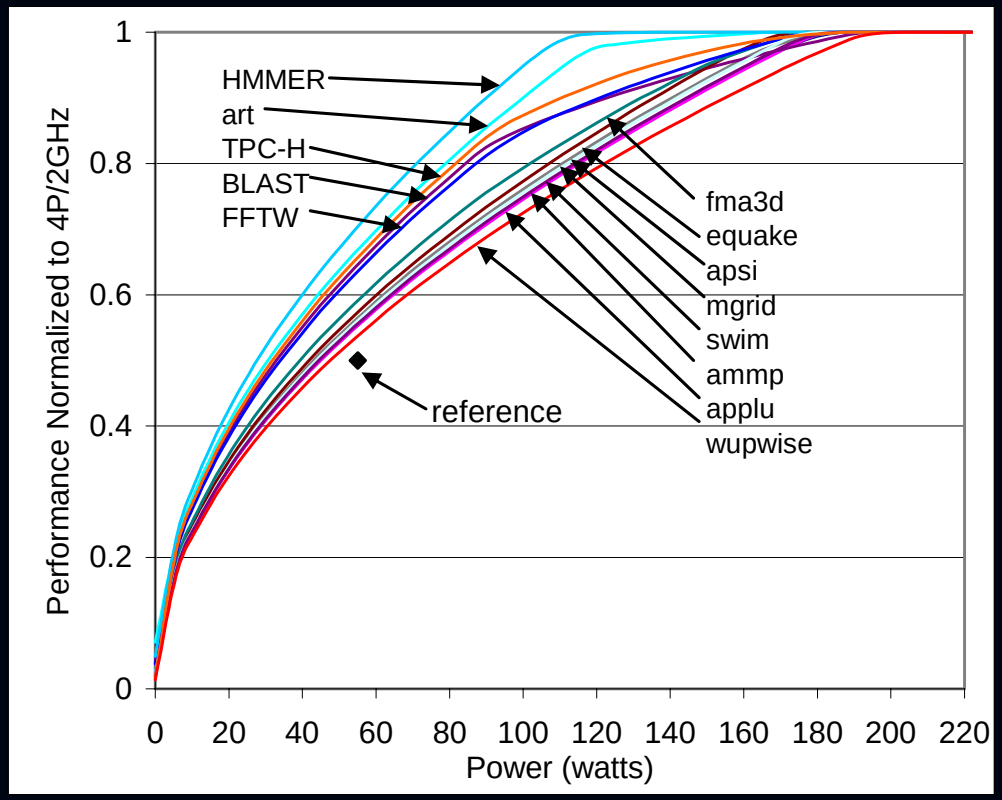


- Current probe on +12V input wires to voltage regulators
- Multimeter readings transferred to a client PC
- 600 current samples/second
- 20,000 samples to 400,000 samples for each benchmark

Throttle Simulator

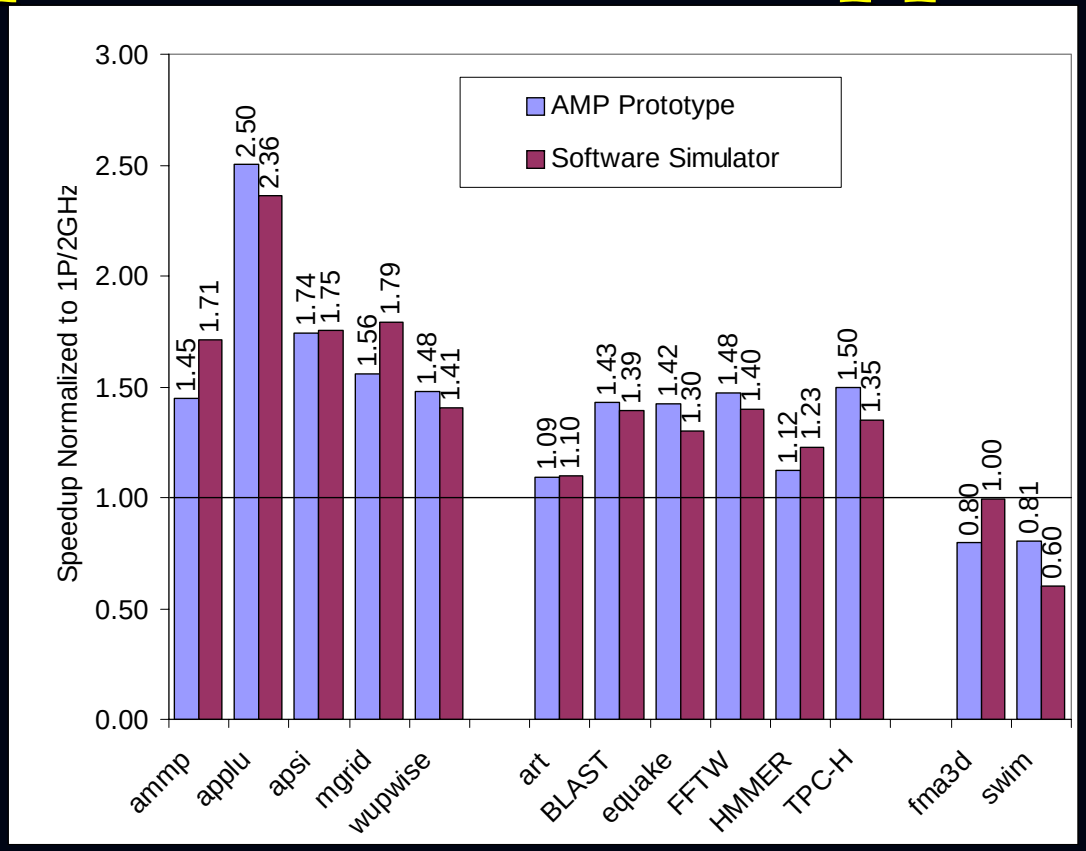
- Read trace of supply current
- Simulate EPI throttle that regulates all processors uniformly
- Output execution time
- Programmable
  - Power threshold
  - Feedback loop gain constant
  - Power, performance relationship

Simulator Results



- Reference point: 0.5 performance at 0.25 power (55 watts)
- Art, BLAST, FFTW, HMMER and TPC-H show least degradation with reduced power
  - Phases of execution where the four CPUs are underutilized.
- Wupwise and applu show most performance degradation as CPU power is constrained

Comparison of Two Approaches



- Two approaches provide comparable results
- AMP prototype introduces thread migration overhead
- Processors almost always run at less than maximum power
- Simulator uniformly slows down processors, memory, and I/O

Conclusion

- Evaluated EPI throttling using
  - AMP prototype
  - Current measurement and software simulation
- EPI throttling gives 38% performance increase
  - Comparing AMP to 4-way SMP
  - Constant power budget
- Mitigated effects of Amdahl's law
  - Run sequential phase on high EPI processor
  - Run parallel phases on multiple low EPI processors
- EPI throttling is inevitable

Areas for Future Work

- How well does EPI throttling scale with future large CMPs?
- What percentage of a typical software workload is comprised of an inherently sequential portion?
- What is the best microarchitecture for an EPI-throttled CMP?
- What are the software implications of the EPI throttle?
- How should an EPI throttle function given multiple, potentially conflicting goals?