Amdahl’s Law in the Multicore Era

Mark D. Hill and Michael R. Marty

University of Wisconsin—Madison
August 2008 @ Semiahmoo Workshop

IBM’s Dr. Thomas Puzak:
Everyone knows Amdahl’s Law
But quickly forgets it!
Executive Summary

- Develop A Corollary to Amdahl’s Law
  - Simple Model of Multicore Hardware
  - Complements Amdahl’s software model
  - Fixed chip resources for cores
  - Core performance improves sub-linearly with resources

- Research Implications
  1. Need Dramatic Increases in Parallelism (No Surprise)
     a. 99% parallel limits 256 cores to speedup 72
     b. New Moore’s Law: Double Parallelism Every Two Years?
  2. Many larger chips need increased core performance
  3. HW/SW for asymmetric designs (one/few cores enhanced)
  4. HW/SW for dynamic designs (serial $\rightarrow$ parallel)
Outline

• Multicore Motivation & Research Paper Trends
• Recall Amdahl’s Law
• A Model of Multicore Hardware
  • Symmetric Multicore Chips
  • Asymmetric Multicore Chips
  • Dynamic Multicore Chips
• Caveats & Wrap Up
How has Architecture Research Prepared?


8/6/2008

Wisconsin Multifacet Project
How has Architecture Research Prepared?

Will Architecture Research Overreact?

Source: Hill, 2/2008
What About PL/Compilers (PLDI) Research?

Percent Multiprocessor Papers

PLDI Begins

End of Small SMP Bulge?

Lead up to Multicore

Gentle Multicore Ramp

What Next?

Source: Steve Jackson, 3/2008
What About Systems (SOSP/OSDI) Research?

Percent Multiprocessor Papers

What Next?

NO Multicore Ramp (Yet)

Lead up to Multicore

Small SMP Bulge

Source: Michael Swift, 3/2008

SOSP odd years only → ODSI even & SOSP odd
Outline

• Multicore Motivation & Research Paper Trends

• Recall Amdahl’s Law

• A Model of Multicore Hardware

• Symmetric Multicore Chips

• Asymmetric Multicore Chips

• Dynamic Multicore Chips

• Caveats & Wrap Up
Recall Amdahl’s Law

• Begins with Simple Software Assumption (Limit Arg.)
  – Fraction $F$ of execution time perfectly parallelizable
  – No Overhead for
    – Scheduling
    – Communication
    – Synchronization, etc.
  – Fraction $1 - F$ Completely Serial

• Time on 1 core = \( \frac{1 - F}{1} + \frac{F}{1} = 1 \)

• Time on $N$ cores = \( \frac{1 - F}{1} + \frac{F}{N} \)
Recall Amdahl’s Law [1967]

\[
\text{Amdahl’s Speedup} = \frac{1}{1 - F} + \frac{F}{N}
\]

- For mainframes, Amdahl expected $1 - F = 35\%$
  - For a 4-processor speedup = 2
  - For infinite-processor speedup < 3
  - Therefore, stay with mainframes with one/few processors

- Amdahl’s Law applied to Minicomputer to PC Eras
- What about the Multicore Era?
Designing Multicore Chips Hard

• Designers must confront single-core design options
  – Instruction fetch, wakeup, select
  – Execution unit configuration & operand bypass
  – Load/queue(s) & data cache
  – Checkpoint, log, runahead, commit.

• As well as additional design degrees of freedom
  – How many cores? How big each?
  – Shared caches: levels? How many banks?
  – Memory interface: How many banks?
  – On-chip interconnect: bus, switched, ordered?
Want Simple Multicore Hardware Model

To Complement Amdahl’s Simple Software Model

(1) Chip Hardware Roughly Partitioned into
   - Multiple Cores (with L1 caches)
   - The Rest (L2/L3 cache banks, interconnect, pads, etc.)
   - Changing Core Size/Number does NOT change The Rest

(2) Resources for Multiple Cores Bounded
   - Bound of \( N \) resources per chip for cores
   - Due to area, power, cost ($$$), or multiple factors
   - Bound = Power? (but our pictures use Area)
(3) Micro-architects can improve single-core performance using more of the bounded resource

- **A Simple Base Core**
  - Consumes 1 Base Core Equivalent (BCE) resources
  - Provides performance normalized to 1

- **An Enhanced Core (in same process generation)**
  - Consumes $R$ BCEs
  - Performance as a function $\text{Perf}(R)$

- What does function $\text{Perf}(R)$ look like?
More on Enhanced Cores

- (Performance Perf(R) consuming R BCEs resources)
- If Perf(R) > R ➔ Always enhance core
- Cost-effectively speedups both sequential & parallel
- Therefore, Equations Assume Perf(R) < R
  - Graphs Assume Perf(R) = Square Root of R
    - 2x performance for 4 BCEs, 3x for 9 BCEs, etc.
    - Why? Models diminishing returns with “no coefficients”
    - Alpha EV4/5/6 [Kumar 11/2005] & Intel’s Pollack’s Law

- How to speedup enhanced core?
  - <Insert favorite or TBD micro-architectural ideas here>
Outline

• Multicore Motivation & Research Paper Trends
• Recall Amdahl’s Law
• A Model of Multicore Hardware

• **Symmetric Multicore Chips**
• Asymmetric Multicore Chips
• Dynamic Multicore Chips
• Caveats & Wrap Up
How Many (Symmetric) Cores per Chip?

- Each Chip Bounded to N BCEs (for all cores)
- Each Core consumes R BCEs
- Assume Symmetric Multicore = All Cores Identical
- Therefore, $N/R$ Cores per Chip — $(N/R)R = N$
- For an $N = 16$ BCE Chip:
Performance of Symmetric Multicore Chips

• Serial Fraction 1-F uses 1 core at rate Perf(R)
  Serial time = (1 – F) / Perf(R)

• Parallel Fraction uses N/R cores at rate Perf(R) each
  Parallel time = F / (Perf(R) * (N/R)) = F*R / Perf(R)*N

• Therefore, w.r.t. one base core:
  \[
  \text{Symmetric Speedup} = \frac{1}{1 - F \frac{1}{\text{Perf}(R)}} + \frac{F \frac{R}{\text{Perf}(R)*N}}{\text{Enhanced Cores speed Serial & Parallel}}
  \]

• Implications?
Symmetric Multicore Chip, N = 16 BCEs

F=0.5, Opt. Speedup $S = 4 = \frac{1}{0.5/4 + 0.5*16/(4*16)}$

Need to increase parallelism to make multicore optimal!
Symmetric Multicore Chip, N = 16 BCEs

At F=0.9, Multicore optimal, but speedup limited

Need to obtain even more parallelism!
Symmetric Multicore Chip, N = 16 BCEs

F matters: Amdahl’s Law applies to multicore chips
MANY Researchers should target parallelism F first
• **Technologist’s Moore’s Law**
  – Double Transistors per Chip every 2 years
  – Slows or stops: TBD

• **Microarchitect’s Moore’s Law**
  – Double Performance per Core every 2 years
  – Slowed or stopped: Early 2000s

• **Multicore’s Moore’s Law**
  – Double Cores per Chip every 2 years
  – & Double Parallelism per Workload every 2 years
  – & Aided by Architectural Support for Parallelism
  – = Double Performance per Chip every 2 years
  – Starting now

• Software as Producer, not Consumer, of Performance Gains!
As Moore’s Law enables $N$ to go from 16 to 256 BCEs, More cores? Enhance cores? Or both?

Recall $F=0.9$, $R=2$, Cores=8, **Speedup=6.7**
Symmetric Multicore Chip, $N = 256$ BCEs

As Moore’s Law increases $N$, often need enhanced core designs

Some arch. researchers should target single-core performance

MORE CORES & ENHANCE CORES!

8/6/2008 Wisconsin Multifacet Project
Software for Large Symmetric Multicore Chips

• F matters: Amdahl’s Law applies to multicore chips

  • $N = 256$
    \[- F=0.9 \implies \text{Speedup} = 27 \at R = 28 \]
    \[- F=0.99 \implies \text{Speedup} = 80 \at R = 3 \]
    \[- F=0.999 \implies \text{Speedup} = 204 \at R = 1 \]

  • $N = 1024$
    \[- F=0.9 \implies \text{Speedup} = 53 \at R = 114 \]
    \[- F=0.99 \implies \text{Speedup} = 161 \at R = 10 \]
    \[- F=0.999 \implies \text{Speedup} = 506 \at R = 1 \]

• Researchers must target parallelism $F$ first
Aside: Cost-Effective Parallel Computing

• Isn’t Speedup(C) < C Inefficient? (C = #cores)

• Much of a Computer’s Cost OUTSIDE Processor

• Let Costup(C) = Cost(C)/Cost(1)

• Parallel Computing Cost-Effective:
  Speedup(C) > Costup(C)

• 1995 SGI PowerChallenge w/ 500MB:
  Costup(32) = 8.6

Multicores have even lower Costups!!!
Outline

• Multicore Motivation & Research Paper Trends
• Recall Amdahl’s Law
• A Model of Multicore Hardware
• Symmetric Multicore Chips
  • Asymmetric Multicore Chips
• Dynamic Multicore Chips
• Caveats & Wrap Up
Asymmetric (Heterogeneous) Multicore Chips

• Symmetric Multicore Required All Cores Equal
• Why Not Enhance Some (But Not All) Cores?
  – For Amdahl’s Simple Software Assumptions
    – One Enhanced Core
    – Others are Base Cores
• How?
  – <fill in favorite micro-architecture techniques here>
  – Model ignores design cost of asymmetric design

• How does this effect our hardware model?
How Many Cores per Asymmetric Chip?

- Each Chip Bounded to N BCEs (for all cores)
- One R-BCE Core leaves N-R BCEs
- Use N-R BCEs for N-R Base Cores
- Therefore, 1 + N - R Cores per Chip
- For an N = 16 BCE Chip:

  **Symmetric:** Four 4-BCE cores
  **Asymmetric:** One 4-BCE core & Twelve 1-BCE base cores
Performance of Asymmetric Multicore Chips

• Serial Fraction 1-F same, so time = \((1 - F) / \text{Perf}(R)\)

• Parallel Fraction F
  – One core at rate \(\text{Perf}(R)\)
  – \(N-R\) cores at rate 1
  – Parallel time = \(F / (\text{Perf}(R) + N - R)\)

Therefore, w.r.t. one base core:

\[
\text{Asymmetric Speedup} = \frac{1}{1 - \frac{F}{\text{Perf}(R)}} + \frac{F}{\text{Perf}(R) + N - R}
\]
Asymmetric Multicore Chip, N = 256 BCEs

Number of Cores = 1 (Enhanced) + 256 – R (Base)

How do Asymmetric & Symmetric speedups compare?
Recall Symmetric Multicore Chip, N = 256 BCEs

Recall $F=0.9$, $R=28$, Cores=9, Speedup=26.7
Asymmetric Multicore Chip, N = 256 BCEs

Asymmetric offers greater speedups potential than Symmetric

In Paper: As Moore’s Law increases N, Asymmetric gets better

Some arch. researchers should target asymmetric multicores
Asymmetric Multicore: 3 Software Issues

1. Schedule computation (e.g., when to use bigger core)
2. Manage locality (e.g., sending code or data can sap gains)
3. Synchronize (e.g., asymmetric cores reaching a barrier)

At What Level?
- Application Programmer
- Library Author
- Compiler
- Runtime System
- Operating System
- Hypervisor (Virtual Machine Monitor)
- Hardware

More Info (?)
More Leverage (?)
Outline

• Multicore Motivation & Research Paper Trends
• Recall Amdahl’s Law
• A Model of Multicore Hardware
• Symmetric Multicore Chips
• Asymmetric Multicore Chips
• Dynamic Multicore Chips
• Caveats & Wrap Up
Dynamic Multicore Chips, Take 1

• Why NOT Have Your Cake and Eat It Too?

• N Base Cores for Best Parallel Performance

• Harness R Cores Together for Serial Performance

• How? DYNAMICALLY Harness Cores Together
  – <insert favorite or TBD techniques here>
Dynamic Multicore Chips, Take 2

- Let POWER provide the limit of N BCEs
- While Area is Unconstrained (to first order)

Result: N base cores for parallel; large core for serial
- [Chakraborty, Wells, & Sohi, Wisconsin CS-TR-2007-1607]
- When Simultaneous Active Fraction (SAF) < ½
Performance of Dynamic Multicore Chips

- N Base Cores with R BCEs used Serially
- Serial Fraction 1-F uses R BCEs at rate Perf(R)
  \[ \text{Serial time} = \frac{1 - F}{\text{Perf}(R)} \]
- Parallel Fraction F uses N base cores at rate 1 each
  \[ \text{Parallel time} = \frac{F}{N} \]
- Therefore, w.r.t. one base core:
  \[ \text{Dynamic Speedup} = \frac{1}{\frac{1 - F}{\text{Perf}(R)} + \frac{F}{N}} \]
Recall Asymmetric Multicore Chip, N = 256 BCEs

What happens with a dynamic chip?

F=0.99
R=41
Cores=216
Speedup=166
Dynamic multicore chip, N = 256 BCEs

Dynamic offers greater speedup potential than Asymmetric Arch. Researchers should target dynamically harnessing cores.

$F=0.99$  
$R=256$ (vs. 41)  
Cores=256 (vs. 216)  
Speedup=223 (vs. 166)
Asymmetric Multicore: 3 Software Issues

1. Schedule computation (e.g., when to use bigger core)
2. Manage locality (e.g., sending code or data can sap gains)
3. Synchronize (e.g., asymmetric cores reaching a barrier)

At What Level?
- Application Programmer
- Library Author
- Compiler
- Runtime System
- Operating System
- Hypervisor (Virtual Machine Monitor)
- Hardware

Dynamic Challenges > Asymmetric Ones
Dynamic chips due to power likely
Outline

• Multicore Motivation & Research Paper Trends
• Recall Amdahl’s Law
• A Model of Multicore Hardware
• Symmetric Multicore Chips
• Asymmetric Multicore Chips
• Dynamic Multicore Chips

• Caveats & Wrap Up
Three Multicore Amdahl’s Law

**Symmetric Speedup**

\[
\text{Sequential Section} = \frac{1}{1 - \frac{F}{\text{Perf}(R)} + \frac{F}{\text{Perf}(R) \times N}} + \frac{F}{\text{Perf}(R) \times N}
\]

**Asymmetric Speedup**

\[
\text{Sequential Section} = \frac{1}{1 - \frac{F}{\text{Perf}(R)} + \frac{F}{\text{Perf}(R) \times (N - R)}} + \frac{F}{\text{Perf}(R) \times (N - R)}
\]

**Dynamic Speedup**

\[
\text{Sequential Section} = \frac{1}{1 - \frac{F}{\text{Perf}(R)} + \frac{F}{N}} + \frac{F}{N}
\]
Software Model Charges 1 of 2

• Serial fraction not totally serial
• Can extend software model to tree algorithms, etc.

• Parallel fraction not totally parallel
• Can extend for varying or bounded parallelism

• Serial/Parallel fraction may change
• Can extend for Weak Scaling [Gustafson, CACM’88]
• Run larger, more parallel problem in constant time
• But prudent architectures support Strong Scaling
Software Model Charges 2 of 2

- Synchronization, communication, scheduling effects?
- Can extend for overheads and imbalance

- Software challenges for asymmetric multicore worse
  Can extend for asymmetric scheduling, etc.

- Software challenges for dynamic multicore greater
  Can extend to model overheads to facilitate

- Future software will be totally parallel (see “my work”)
- I’m skeptical; not even true for MapReduce
Hardware Model Charges 1 of 2

- Naïve to consider total resources for cores fixed
- Can extend hardware model to how core changes effect The Rest

- Naïve to bound Cores by one resource (esp. area)
- Can extend for Pareto optimal mix of area, dynamic/static power, complexity, reliability, …

- Naïve to ignore challenges due to off-chip bandwidth limits & benefits of last-level caching
- Can extend for modeling these
Naïve to use performance = square root of resources
Can extend as equations can use any function

We architects can’t scale Perf(R) for very large R
True, not yet.

We architects can’t dynamically harness very large R
True, not yet

So what should computer scientists do about it?
Three-Part Charge

Architects: Build more-effective multicore hardware
• Don’t lament that we can’t do, but do it!

Computer Scientists: Implement “3rd Moore’s Law”
• Double Parallelism Every Two Years
• Consider Symmetric, Asymmetric, & Dynamic Chips

Finally, We must all work together
• Keep (cost-) performance gains progressing
• Parallel Programming & Parallel Computers
Dynamic Multicore Chip, $N = 1024$ BCEs

NOT Possible Today

NOT Possible EVER Unless We Dream & Act

8/6/2008

Wisconsin Multifacet Project
Executive Summary

• Develop A Corollary to Amdahl’s Law
  – Simple Model of Multicore Hardware
  – Complements Amdahl’s software model
  – Fixed chip resources for cores
  – Core performance improves sub-linearly with resources

• Research Implications
  (1) Need Dramatic Increases in Parallelism (No Surprise)
      • 99% parallel limits 256 cores to speedup 72
  (2) New Moore’s Law: Double Parallelism Every Two Years?
  (3) Many larger chips need increased core performance
  (4) HW/SW for asymmetric designs (one/few cores enhanced)
  (5) HW/SW for dynamic designs (serial ↔ parallel)
Symmetric Multicore Chip, N = 16 BCEs

Symmetric Speedup vs. R BCEs for different F values:

- F = 0.999
- F = 0.99
- F = 0.975
- F = 0.9
- F = 0.5
Symmetric Multicore Chip, N = 256 BCEs

Symmetric Speedup

R BCEs

F=0.999
F=0.99
F=0.975
F=0.9
F=0.5
Symmetric Multicore Chip, N = 1024 BCEs

The graph shows the symmetric speedup as a function of the number of BCEs (Basic Computational Elements) for different values of parameter $F$. The speedup decreases as the number of BCEs increases, and the curves diverge with different values of $F$: $F=0.999$, $F=0.99$, $F=0.975$, $F=0.9$, and $F=0.5$. The graph indicates that the speedup is significantly higher with lower values of $F$, especially for small numbers of BCEs.
Asymmetric Multicore Chip, N = 16 BCEs
Asymmetric Multicore Chip, N = 256 BCEs

Asymmetric Speedup vs R BCEs for different values of F:
- F = 0.999
- F = 0.99
- F = 0.975
- F = 0.9
- F = 0.5
Asymmetric Multicore Chip, N = 1024 BCEs

Asymmetric Speedup

R BCEs

F=0.999
F=0.99
F=0.975
F=0.9
F=0.5
Dynamic Multicore Chip, N = 16 BCEs

Dynamic Speedup

R BCEs

$F=0.999$

$F=0.99$

$F=0.975$

$F=0.9$

$F=0.5$
Dynamic Multicore Chip, N = 256 BCEs

Dynamic Speedup vs. R BCEs for different values of F:
- F = 0.999
- F = 0.99
- F = 0.975
- F = 0.9
- F = 0.5

8/6/2008 Wisconsin Multifacet Project
Dynamic Multicore Chip, N = 1024 BCEs

Dynamic Speedup

R BCEs

F=0.999

F=0.99

F=0.975

F=0.9

F=0.5