Reevaluating Amdahl’s Law in the Multicore Era

Xian-He Sun

Illinois Institute of Technology
Chicago, Illinois
sun@iit.edu
# TOP 10 Machines (6/2004)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>Computer</th>
<th>#proc</th>
<th>TF/s</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Earth simulator center</td>
<td>Earth simulator/NEC</td>
<td>5120</td>
<td>35860</td>
<td>Japan</td>
</tr>
<tr>
<td>2</td>
<td>LLNL</td>
<td>Thunder/Intel Itanium 2 Tiger 41.4GHz Quadrics</td>
<td>4096</td>
<td>19940</td>
<td>USA</td>
</tr>
<tr>
<td>3</td>
<td>LANL</td>
<td>ASCI Q/AlphaServer SC45 1.25GHz</td>
<td>8192</td>
<td>13880</td>
<td>USA</td>
</tr>
<tr>
<td>4</td>
<td>IBM-Rochester</td>
<td>BlueGene/LDD1 Prototype</td>
<td>8192</td>
<td>11680</td>
<td>USA</td>
</tr>
<tr>
<td>5</td>
<td>NCAS</td>
<td>Tungsten/PowerEdge 1750, P4 Xeon 3.06Ghz</td>
<td>2500</td>
<td>9819</td>
<td>USA</td>
</tr>
<tr>
<td>6</td>
<td>ECMWF</td>
<td>eServer P Series690 IBM</td>
<td>2112</td>
<td>8955</td>
<td>UK</td>
</tr>
<tr>
<td>7</td>
<td>RIKEN</td>
<td>RIKEN Super Combined Cluster/Fujitsu</td>
<td>2048</td>
<td>8728</td>
<td>Japan</td>
</tr>
<tr>
<td>8</td>
<td>IBM-Thomas Watson</td>
<td>BlueGene/LDD2 Prototype</td>
<td>2096</td>
<td>8655</td>
<td>USA</td>
</tr>
<tr>
<td>9</td>
<td>PNNL</td>
<td>Integrity rx2600 Itanium 21.5 GHz</td>
<td>1936</td>
<td>8633</td>
<td>USA</td>
</tr>
<tr>
<td>10</td>
<td>Shanghai Supercomputer Center</td>
<td>Dawning 4000A Opteron 2.2GHz</td>
<td>2560</td>
<td>8061</td>
<td>China</td>
</tr>
</tbody>
</table>

[http://www.top500.org/list/2004/06/]
Cloud: Integrated Resource

Provide virtual computing environments on demand
Scalable Computing: the Way to High-performance

IBM BG/P

Rack

32 Node Cards
1024 chips, 4096 procs

Cabled 8x8x16

Node Board
(32 chips 4x4x2)
32 compute, 0-2 IO cards

Compute Card
1 chip, 20 DRAMs

Chip
4 cores

850 MHz
8 MB EDRAM

13.6 GF/s
2.0 GB DDR
Supports 4-way SMP

435 GF/s
64 GB

14 TF/s
2 TB

1 PF/s
144 TB

Maximum System
256 racks
3.5 PF/s
512 TB

Front End Node / Service Node
System p Servers
Linux SLES10

HPC SW:
Compilers
GPFS
ESSL
Loadleveler

Source: ANL ALCF

10/29/2009 Scalable Computing Software Lab, Illinois Institute of Technology
Multicore Adds in Another Dimension

IBM Cell: 8 slave cores + 1 master core, 2005

Sun T2: 8 cores, 2007

AMD Phenom: 4 cores, 2007

Intel Dunnington: 6 cores, 2008
No in the Mood to Scale Up, yet

Intel Dunnington: 6 cores, 2009

Sun UltraSPARC Rock: 16 cores, 2009

AMD Opteron “Istanbul”: 6 Cores, 2009

IBM Power-7: 8 cores, 2010
Why not Scale up the Number of Cores?

Perception/technology?

With a Staggering Potential

Assume we scale entire current single core chip & replicate to fill 280 sq mm die.
Whereas Technology is Available

Kilocore: 256-core prototype
By Rapport Inc.

Tesla C1060:
240 cores, by NVDIA

GRAPE-DR chip:
512-core, By Japan

Quadro FX 5800: 240 cores,
By NVDIA.

NVIDIA Fermi: 512 CUDA cores

GRAPE-DR testboard
It All Starts with Amdahl’s Law

- Amdahl’s law (Amdahl’s speedup model)

\[
Speedup_{Amdahl} = \frac{1}{(1 - f) + \frac{f}{n}}
\]

\[
\lim_{{n \to \infty}} Speedup_{Amdahl} = \frac{1}{1 - f}
\]

- \( f \) is the parallel portion
- Implications
Amdahl’s Law for Multicore (Hill&Marty)

- Study the limitation of multicore architecture based on Amdahl’s law for parallel processing and hardware concern
  - $n$ BCEs (Base Core Equivalents)
  - A powerful $\text{perf}(r)$ core built with $r$ BCEs is best choice from design concern

![Symmetric and Asymmetric Multicore Architectures](image-url)
Amdahl’s Law for Multicore (Hill&Marty)

- Speedup of symmetric architecture
  \[ Speedup_{symmetric}(f,n,r) = \frac{1}{1 - f} + \frac{f \cdot r}{\text{perf}(r) \cdot n} \]

- Speedup of asymmetric architecture
  \[ Speedup_{asymmetric}(f,n,r) = \frac{1}{1 - f} + \frac{f}{\text{perf}(r) + n - r} \]
History Repeats Itself (back to 1988)?

All have up to 8 processors, citing Amdahl’s law,
\[ \lim_{n \to \infty} \text{Speedup}_{\text{Amdahl}} = \frac{1}{1 - f} \]

IBM 7030 Stretch

IBM 7950 Harvest

Cray X-MP
Fastest computer 1983-1985

Cray Y-MP
Terms of Scalable Computing (today)

LANL Roadrunner:
18,360 processors, 130,464 cores
2009 World’s fastest supercomputer

TACC Ranger:
15,744 processors, 2008

The scale size is far beyond implication of Amdahl’s law

ANL Intrepid:
20,480 processors, 2008
Scalable Computing

- Tacit assumption in Amdahl’s law
  - The problem size is fixed
  - The speedup emphasizes time reduction
- Gustafson’s Law, 1988
  - Fixed-time speedup model
    \[
    \text{Speedup}_{\text{fixed-time}} = \frac{\text{Sequential Time of Solving Scaled Workload}}{\text{Parallel Time of Solving Scaled Workload}} = (1 - f) + nf
    \]
- Sun and Ni’s law, 1990
  - Memory-bounded speedup model
    \[
    \text{Speedup}_{\text{memory-bounded}} = \frac{\text{Sequential Time of Solving Scaled Workload}}{\text{Parallel Time of Solving Scaled Workload}} = \frac{(1 - f) + fG(n)}{(1 - f) + fG(n) / n}
    \]
Revisit Amdahl’s Law for Multicore

\[ \text{Speedup}_{\text{fixed-size}} = \frac{T_{\text{original}}}{T_{\text{enhanced}}} \]

\[ T_{\text{original}} = \frac{w}{\text{perf}(1)} = w \]

\[ T_{\text{enhanced}} = \frac{(1 - f)w}{\text{perf}(r)} + \frac{fw}{n \cdot \text{perf}(r)} \]

\[ \text{Speedup}_{\text{fixed-size}} = \frac{w}{\text{perf}(1)} \]

\[ = \frac{1}{\frac{1 - f}{\text{perf}(r)} + \frac{f \cdot r}{\text{perf}(r) \cdot n}} \]

Hill and Marty’s findings
Fixed-time Model for Multicore

- Emphasis on work finished in a fixed time
- Problem size is scaled from $w$ to $w'$
- $w'$: Work finished within the fixed time, when the number of cores scales from $r$ to $mr$

$$\frac{(1-f)w}{\text{perf}(r)} + \frac{fw}{\text{perf}(r)} = \frac{(1-f)w}{\text{perf}(r)} + \frac{fw'}{\text{perf}(r)m} \Rightarrow w' = mw$$

- The scaled fixed-time speedup

$$\text{Speedup}_{\text{fixed-time}} = \frac{\text{Time of Solving } w' \text{ in Original Mode}}{\text{Time of Solving } w \text{ in Original Mode}}$$

$$= \frac{(1-f)w}{\text{perf}(r)} + \frac{fw'}{\text{perf}(r)} = \frac{w}{\text{perf}(r)} = (1-f) + mf$$
Fixed-time Speedup for Multicore

Scales linearly
Memory-bounded Model for Multicore

- Problem size is scaled from \( w \) to \( w^* \)
- \( w^* \): Work executed under memory limitation (each core has its own L1 cache)
- \( w^* = g(m)w \), where \( g(m) \) is the increased workload as the memory capacity increases \( m \) times \( (g(m) = 0.38m^{3/2}, \text{ for matrix-multiplication } 2N^3 \text{ v.s. } 3N^2) \)
- The scaled memory-bounded speedup

\[
Speedup_{memory-bounded} = \frac{\text{Time of Solving } w^* \text{ in Original Mode}}{\text{Time of Solving } w \text{ in Original Mode}}
\]

\[
= \frac{(1 - f) + g(m)f}{(1 - f) + \frac{g(m)f}{m}}
\]
Memory-bounded Speedup of Multicore Architecture

\[ g(m) = 0.38m^{3/2} \]  
for matrix-multiplication 2N3 v.s. 3N2

Scales linearly and better than fixed-time
Perspective: a comparison

Fixed-size, Fixed-time and Memory-bounded Speedup of Multicore Architecture

Scalable computing shows an optimistic view
Result and Implications

- **Result**: The scalable computing concept and the two scaled speedup models are applicable to multicore architecture.
- **Implication 1**: Amdahl’s law (Hill&Marty) presents a limited and pessimistic view.
- **Implication 2**: Multicore is scalable in term of the number of cores.
- **Implication 3**: The memory-bounded model reveals the relation between scalability and memory capacity and requirement.

**Question:**

Is data access the actual performance constraint of multicore architecture?
Processor-memory performance gap

- Processor performance increases rapidly
  - Uni-processor: ~52% until 2004, ~25% since then
  - New trend: multi-core/many-core architecture
    - Intel TeraFlops chip, 2007
  - Aggregate processor performance much higher
- Memory: ~9% per year
- Processor-memory speed gap keeps increasing
Multicore Scalability Analysis

- **Architecture**
  - $N$ cores
  - Data contention to L2
  - Increase cores does not improve data access speed
Application: Iterative Solvers

- Two phases:
  - Computing phase and communication phase

Dense Solver
$k^3$ comp, $k^2$ memory
Data Access as the Scalability Constraint

- Phased computing model (embarrassing parallel, meta-tasks)
- Assume a task has two parts, $w = w_p + w_c$
  - Data processing work, $w_p$
  - Data communication (access) work, $w_c$
- Fixed-size speedup with data-access processing consideration

$$Speedup = \frac{1}{\frac{w_c}{perf(r)} + \frac{w_p \cdot r}{perf(r) \cdot n}}$$
Scaled Speedup under Memory-wall

- Assuming data access time is fixed, Fixed-time model constraint

\[
\frac{w_c}{\text{perf}(r)} + \frac{w_p}{\text{perf}(r)} = \frac{w_c}{\text{perf}(r)} + \frac{w_p'}{m \cdot \text{perf}(r)} \Rightarrow w_p' = mw_p
\]

- Fixed-time speedup

\[
\frac{w_c}{\text{perf}(r)} + \frac{w_p'}{m \cdot \text{perf}(r)} = \frac{w_c + m \cdot w_p}{w_c + w_p} = (1 - f') + mf'
\]

\[
f' = \frac{w_p}{w_c + w_p}
\]

- Memory-bounded speedup

- With \( g(m) = 0.38m^{3/2} \) memory-bounded speedup is bigger than fixed-time speedup

- \( g(m) \) equals one, memory-bounded is the as fixed-size, \( g(m) \) equals \( m \), then memory-bound is the same as fixed-time
Mitigating Memory-wall Effect

- **Result:** Multicore is scalable, but under the assumption
  - Data access time is fixed and does not increase with the amount of work and the number of cores
- **Implication:** Data access is the bottleneck needs attention

- **Data Prefetching**
  - Software prefetching technique
    - Adaptive, compete for computing power, and costly
  - Hardware prefetching technique
    - Fixed, simple, and less powerful

- **Our Solutions**
  - Data Access History Cache (DAHC)
  - Server-based Push Prefetching
Hybrid Adaptive Prefetching Architecture
Data Access History Cache: a hardware solution for memory
Push-IO: A Software Solution for I/O
Dynamic Application-specific I/O Optimization Architecture
Conclusion

- Cloud computing and multicore/manycore architecture lead to the future of computing
- Multicore architecture is scalable
- Scaling up the number of cores can continually improve performance, if the data access delay is fixed
- Data access is the killing factor of performance
- Mitigating memory-wall: Data prefetching
  - Data Access History Cache (DAHC)
  - Server-based Push Prefetching
- Mitigating memory-wall: Application-specific data access system
Thank you!

To visit http://www.cs.iit.edu/~scs