CS 250B: Modern Computer Systems

The End of Conventional Performance Scaling

Sang-Woo Jun
Conventional Performance Scaling

- Traditional model of a computer is simple
  - Single, in-order flow of instructions on a processor
  - Simple, in-order memory model

- Large part of computer architecture research involved maintaining this abstraction while improving performance
  - Transparent caches, Transparent superscalar scheduling, ...
  - Same software runs faster tomorrow
  - (Slow software becomes acceptable tomorrow)

- Driven largely by continuing march of Moore's law
Moore’s Law

- What exactly does it mean?
- What is it that is scaling?
Moore’s Law

- Typically cast as: “Performance doubles every X months”

- Actually closer to: “Number of transistors per unit cost doubles every two years”
Moore’s Law

The complexity for minimum component costs has increased at a rate of roughly a factor of two per year.

[...] 

Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years.

-- Gordon Moore, Electronics, 1965

Why is Moore’s Law conflated with processor performance?
Dennard Scaling: Moore’s Law to Performance

“Power density stays constant as transistors get smaller”
  - Robert H. Dennard, 1974

Intuitively:
  - Smaller transistors → shorter propagation delay → faster frequency
  - Smaller transistors → smaller capacitance → lower voltage

- $Power \propto Capacitance \times Voltage^2 \times Frequency$

Moore’s law → Faster performance @ Constant power!
What happened?
(Slightly) More Accurate Processor Power Consumption

Power = \((\text{Active Transistors} \times \text{Capacitance} \times \text{Voltage}^2 \times \text{Frequency})\)  

\[\text{Dynamic power}\]

\[\text{Static power}\]

\[\text{Leakage} \propto \frac{1}{e^{\text{Voltage}}}\]

Unfortunately...

Gate-oxide stopped scaling
Stopped scaling due to leakage
EXTREMELY simplified model!
Power Consumption of High-Density Circuits

- Total power consumption with constant frequency

https://www.design-reuse.com/articles/20296/power-management-leakage-control-process-compensation.html
End of Dennard Scaling

- Even with smaller transistors, we cannot continue reducing power
  - What do we do now?

- Option 1: Continue scaling frequency at increased power budget
  - Chip quickly become too hot to cool!
  - Thermal runaway:
    - Hotter chip $\rightarrow$ increased resistance $\rightarrow$ hotter chip $\rightarrow$ ...
Option 1: Continue Scaling Frequency at Increased Power Budget
Option 2: Stop Frequency Scaling

Dennard Scaling Ended (~2006)

Looking Back: Change of Predictions

Moore’s Law – The number of transistors on integrated circuit chips (1971-2016)
Moore’s law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore’s law.
State of Things at This Point (2006)

- Single-thread performance scaling ended
  - Frequency scaling ended (Dennard Scaling)
  - Instruction-level parallelism scaling stalled ... also around 2005

- Moore’s law continues
  - Double transistors every two years
  - What do we do with them?

K. Olukotun, “Intel CPU Trends”
Crisis Averted With Manycores?
Crisis Averted With Manycores?

Source:
International Roadmap for Semiconductors 2007 edition (http://www.itrs.net/)
What Happened?

\[ Power = \text{(Active Transistors} \times \text{Capacitance} \times \text{Voltage}^2 \times \text{Frequency}) + (\text{Voltage} \times \text{Leakage Current}) \]

Dynamic power

- Can’t keep going up
- Gate-oxide stopped scaling
- Stopped scaling due to leakage

Static power

- Stopped scaling due to leakage
- “Utilization Wall”

Regardless of Moore’s Law, a limited amount of gates can be active at a given time.
Where To, From Here?

- The number of active transistors at a given time is limited
  - Left unchecked, we won’t get much performance improvements even with Moore’s law continuing
  - We need to make the best use of those active transistors!
Also, Scaling Size is Becoming More Difficult!

- Processor fabrication technology has always reduced in size
  - As of end of 2020, 7 nm is cutting edge, working towards 5 nm

Image source: Intel
<table>
<thead>
<tr>
<th>Number of Semiconductor Manufacturers with a Cutting Edge Logic Fab</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SiTerra</strong></td>
</tr>
<tr>
<td><strong>Dongbu HiTek</strong></td>
</tr>
<tr>
<td><strong>ADI</strong></td>
</tr>
<tr>
<td><strong>Atmel</strong></td>
</tr>
<tr>
<td><strong>Rohm</strong></td>
</tr>
<tr>
<td><strong>Sanyo</strong></td>
</tr>
<tr>
<td><strong>Mitsubishi</strong></td>
</tr>
<tr>
<td><strong>ON</strong></td>
</tr>
<tr>
<td><strong>Hitachi</strong></td>
</tr>
<tr>
<td><strong>Cypress</strong></td>
</tr>
<tr>
<td><strong>Sony</strong></td>
</tr>
<tr>
<td><strong>Infineon</strong></td>
</tr>
<tr>
<td><strong>Sharp</strong></td>
</tr>
<tr>
<td><strong>Freescale</strong></td>
</tr>
<tr>
<td><strong>TI</strong></td>
</tr>
<tr>
<td><strong>Fujitsu</strong></td>
</tr>
<tr>
<td><strong>TI</strong></td>
</tr>
<tr>
<td><strong>TI</strong></td>
</tr>
<tr>
<td><strong>Panasonic</strong></td>
</tr>
<tr>
<td><strong>Toshiba</strong></td>
</tr>
<tr>
<td><strong>Toshiba</strong></td>
</tr>
<tr>
<td><strong>Renoves (NCL)</strong></td>
</tr>
<tr>
<td><strong>Panasonic</strong></td>
</tr>
<tr>
<td><strong>STMicroelectronics</strong></td>
</tr>
<tr>
<td><strong>STMicroelectronics</strong></td>
</tr>
<tr>
<td><strong>HMC</strong></td>
</tr>
<tr>
<td><strong>HMC</strong></td>
</tr>
<tr>
<td><strong>IBM</strong></td>
</tr>
<tr>
<td><strong>IBM</strong></td>
</tr>
<tr>
<td><strong>SMIC</strong></td>
</tr>
<tr>
<td><strong>SMIC</strong></td>
</tr>
<tr>
<td><strong>AMD</strong></td>
</tr>
<tr>
<td><strong>AMD</strong></td>
</tr>
<tr>
<td><strong>Samsung</strong></td>
</tr>
<tr>
<td><strong>Samsung</strong></td>
</tr>
<tr>
<td><strong>Samsung</strong></td>
</tr>
<tr>
<td><strong>Samsung</strong></td>
</tr>
<tr>
<td><strong>TSMC</strong></td>
</tr>
<tr>
<td><strong>Intel</strong></td>
</tr>
<tr>
<td><strong>Intel</strong></td>
</tr>
<tr>
<td>180 nm</td>
</tr>
<tr>
<td>90 nm</td>
</tr>
<tr>
<td>45 nm/40 nm</td>
</tr>
<tr>
<td>22 nm/20 nm</td>
</tr>
<tr>
<td>10 nm</td>
</tr>
<tr>
<td>5 nm</td>
</tr>
</tbody>
</table>

Only three players left?!
Where To, From Here?

- Potential Solution 1: The software solution
  - Write efficient software to make the efficient use of hardware resources
  - No longer depend entirely on hardware performance scaling
  - “Performance engineering” software, using hardware knowledge
Impact of Software Performance Engineering

- Multiplying two 2048 x 2048 matrices
  - 16 MiB, doesn’t fit in smaller caches
- Machine: Intel i5-7400 @ 3.00GHz

\[
\begin{array}{c}
\text{A} \\
\vdots \\
\end{array} \times 
\begin{array}{c}
\vdots \\
\vdots \\
\end{array} 
\quad \text{VS} 
\begin{array}{c}
\text{A} \\
\vdots \\
\end{array} \times 
\begin{array}{c}
\vdots \\
\vdots \\
\end{array} 
\end{array}
\]

63.19 seconds

10.39 seconds
(6x performance!)

Last year, we measured 42.13x performance improvement just by writing better software
Where To, From Here?

Solution 2: The specialized architectural solution
  - Chip space is now cheap, but power is expensive
  - Stop depending on more complex general-purpose cores
  - Use space to build heterogeneous systems, with compute engines well-suited for each application
Fine-Grained Parallelism of Special-Purpose Circuits

Example -- Calculating gravitational force:

\[
\frac{G \times m_1 \times m_2}{(x_1 - x_2)^2 + (y_1 - y_2)^2}
\]

8 instructions on a CPU, 16 instructions for two calculations, ...

Specialized datapath can be extremely efficient
  - Pipelined implementation can emit one result per cycle
  - Also, no need for general-purpose overhead such as instruction decoding
    - Much more cores can fit on chip
    - Much lower power consumption per unit

A = G \times m_1
B = A \times m_2
C = x_1 - x_2
D = C^2
E = y_1 - y_2
F = E^2
G = D + F
Ret = B / G

Pipeline stages
Typical Energy Efficiency Benefits of Optimized Hardware

Spectrum of Specialized Hardware

More general

Multicore CPU

General-Purpose GPU

More specialized

Field-Programmable Gate Array (FPGA)

Application-Specific Integrated Circuit (ASIC)

Power efficiency
The Bottom Line:
Architecture is No Longer Transparent

- Optimized software requires architecture knowledge
- Special-purpose “accelerators” (GPU, FPGA, ...) programmed explicitly
- Even general-purpose processors implement specialized instructions
  - Single-Instruction Multiple Data (SIMD) instructions such as AVX
  - Special-purpose instructions sets such as AES-NI
Coming Up

- Before we go into newer technologies, let’s first make sure we make good use of what we have
  - SIMD (SSE, AVX), Cache-optimized code, etc
  - “Performance engineering”
- “Our implementation delivers 9.2X the performance (RPS) and 2.8X the system energy efficiency (RPS/watt) of the best-published FPGA-based claims.”
  - Li et. al., Intel, “Architecting to Achieve a Billion Requests Per Second Throughput on a Single Key-Value Store Server Platform,” ISCA 2015
  - Intel software implementation of memcached