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
  - $\text{Power} \propto \text{Capacitance} \times \text{Voltage}^2 \times \text{Frequency}$

Moore’s law → Faster performance @ Constant power!
Single-Core Performance Scaling Projection

What happened?
(Slightly) More Accurate Processor Power Consumption

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

- Dynamic power
- Static power

**Gate-oxide stopped scaling**

**Stopped scaling due to leakage**

**Exremely 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

But Moore’s Law Continues Beyond 2006

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?

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

**Can't keep going up**

Gate-oxide stopped scaling

Stopped scaling due to leakage

**Dynamic power**

**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 2022, 5 nm is cutting edge, working towards 3 nm

Image source: Intel
### Number of Semiconductor Manufacturers with a Cutting Edge Logic Fab

<table>
<thead>
<tr>
<th>Year 2000</th>
<th>Year 2008</th>
<th>Year 2022</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samsung</td>
<td>Samsung</td>
<td>Samsung</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>TSMC</td>
<td>TSMC</td>
<td>TSMC</td>
</tr>
<tr>
<td>STMicroelectronics</td>
<td>STMicroelectronics</td>
<td>STMicroelectronics</td>
</tr>
<tr>
<td>HLMC</td>
<td>HLMC</td>
<td>HLMC</td>
</tr>
<tr>
<td>UMC</td>
<td>UMC</td>
<td>UMC</td>
</tr>
<tr>
<td>AMD</td>
<td>AMD</td>
<td>AMD</td>
</tr>
<tr>
<td>AMD</td>
<td>AMD</td>
<td>AMD</td>
</tr>
<tr>
<td>SMIC</td>
<td>SMIC</td>
<td>SMIC</td>
</tr>
<tr>
<td>TI</td>
<td>TI</td>
<td>TI</td>
</tr>
<tr>
<td>Panasonic</td>
<td>Panasonic</td>
<td>Panasonic</td>
</tr>
<tr>
<td>Cypress</td>
<td>Cypress</td>
<td>Cypress</td>
</tr>
<tr>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
</tr>
<tr>
<td>Infineon</td>
<td>Infineon</td>
<td>Infineon</td>
</tr>
<tr>
<td>Sharp</td>
<td>Sharp</td>
<td>Sharp</td>
</tr>
<tr>
<td>Freescale</td>
<td>Freescale</td>
<td>Freescale</td>
</tr>
<tr>
<td>Renesas</td>
<td>Renesas</td>
<td>Renesas</td>
</tr>
<tr>
<td>Toshiba</td>
<td>Toshiba</td>
<td>Toshiba</td>
</tr>
<tr>
<td>Fujitsu</td>
<td>Fujitsu</td>
<td>Fujitsu</td>
</tr>
<tr>
<td>TI</td>
<td>TI</td>
<td>TI</td>
</tr>
<tr>
<td>Panasonic</td>
<td>Panasonic</td>
<td>Panasonic</td>
</tr>
<tr>
<td>STM</td>
<td>STM</td>
<td>STM</td>
</tr>
<tr>
<td>UMC</td>
<td>UMC</td>
<td>UMC</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>SMIC</td>
<td>SMIC</td>
<td>SMIC</td>
</tr>
<tr>
<td>AMD</td>
<td>AMD</td>
<td>AMD</td>
</tr>
<tr>
<td>Samsung</td>
<td>Samsung</td>
<td>Samsung</td>
</tr>
<tr>
<td>TSMC</td>
<td>TSMC</td>
<td>TSMC</td>
</tr>
<tr>
<td>Intel</td>
<td>Intel</td>
<td>Intel</td>
</tr>
<tr>
<td>180 nm</td>
<td>130 nm</td>
<td>10 nm</td>
</tr>
<tr>
<td>90 nm</td>
<td>65 nm</td>
<td>7 nm</td>
</tr>
<tr>
<td>65 nm</td>
<td>45 nm/40 nm</td>
<td>5 nm</td>
</tr>
<tr>
<td>32 nm/28 nm</td>
<td>22 nm/20 nm</td>
<td></td>
</tr>
<tr>
<td>16 nm/14 nm</td>
<td>10 nm</td>
<td></td>
</tr>
</tbody>
</table>

Only three players left?!
We Can’t Keep Doing What we Used to

- Limited number of transistors, limited clock speed
  - How to make the ABSOLUTE BEST of these resources?

- Timely example: Apple M1 Processor
  - Claims to outperform everyone (per Apple)
  - How?
    - “8-wide decoder” [...] “16 execution units (per core)”
    - “(Estimated) 630-deep out-of-order”
    - “Unified memory architecture”
    - Hardware/software optimized for each other

What do these mean?
Not just apple! (Amazon, Microsoft, EU, ...)

Image source: Apple
We Can’t Keep Doing What we Used to

**AWS Graviton 2:**
64-Core ARM

**European Processor Accelerator (EPAC):**
4-Core RISC-V +
Variable Precision Accelerator +
Stencil and Tensor Accelerator

Image source: Anandtech, “Amazon’s Arm-based Graviton2 Against AMD and Intel: Comparing Cloud Compute”
Image source: TheNextPlatform, “Europe Inches Closer to Native RISC-V Reality”
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

\[
\begin{array}{c|c}
A & B \\
\hline
\vdots & \vdots \\
\end{array}
\quad \times \quad \begin{array}{c|c}
A & \mathbf{B^T} \\
\hline
\vdots & \vdots \\
\end{array}
\]

63.19 seconds  vs  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

\[ A = G \times m_1 \quad C = x_1 - x_2 \quad E = y_1 - y_2 \]
\[ B = A \times m_2 \quad D = C^2 \quad F = E^2 \]
\[ G = D + F \quad \text{Ret} = B / G \]
Typical Energy Efficiency Benefits of Optimized Hardware

Spectrum of Specialized Hardware

- Multicore CPU
- General-Purpose GPU
- 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