CS 152
Computer Architecture and Engineering
Lecture 3 – Metrics

2014-1-28
John Lazzaro
(not a prof - “John” is always OK)

TA: Eric Love

www-inst.eecs.berkeley.edu/~cs152/
Topics for today’s lecture

 Metrics: Estimating the “goodness” of a CPU design ... so that we can redesign the CPU to be “better”.

 Short Break.

 A case study in microcode control: the Motorola 68000, the CPU that powered the original Macintosh. [see Lecture 5 slides for this topic]

 Administrivia: Will announce office hours soon ...
Todd Hamilton,
iWatch concept.
Gray-scale computer graphics model.

Todd Hamilton, iWatch concept.
Todd Hamilton, iWatch concept.
Animated model ...

Then the baton is passed to us.

We use models to do stepwise refinement of the silicon that powers the consumer product.

Todd Hamilton, iWatch concept.
Four metrics:

**Performance**
Execution time of a program.

**Cost**
How many dollars to manufacture.

**Energy**
Joules required to execute a program.

**Time to Market**
Will we ship a product before our competitors?

Today’s Focus

For a later lecture ...

For a later lecture ...

For a later lecture ...
Performance Measurement

(as seen by the customer)
Who (sensibly) upgrades CPUs often?

A professional who turns CPU cycles into money, and who is cycle-limited.

Artist tool: animation, video special effects.
How to decide to buy a new machine?

Measure After Effects “execution time” on a representative render “workload”

“Night flight”
City map and clouds computed “on the fly” with fractals
CPU intensive
Trivial I/O

(still shot from the movie)
Interpreting Execution Time

Performance = \frac{1}{\text{Execution Time}} = 2.85 \text{ renders/hour}

1.5 GHz PB (Y) is \( N \) times faster than 1.25 GHz PB (X). \( N \) is ?

\[
N = \frac{\text{Performance (Y)}}{\text{Performance (X)}} = \frac{\text{Execution Time (X)}}{\text{Execution Time (Y)}} = 1.19
\]

PB 1.5 Gz: 3.4 renders/hour. PB 1.25: 2.85 renders/hour. Might make the difference in meeting a deadline ...
Execution Time: time for one job to complete

Throughput: # of independent jobs/hour completed

Assume G5 MP execution time faster because AE isn’t parallelized on Opteron CPUs.

However, G5 and Opteron may have same throughput.
Performance Measurement
(as seen by a CPU designer)

Q. Why do we care about After Effect’s performance?
A. We want the CPU we are designing to run it well!
Step 1: Analyze the right measurement!

CPU Time:
Time the CPU spends running program under measurement.

Measuring CPU time (Unix):
% time <program name>
25.77u 0.72s 0:29.17 90.8%

Response Time:
Total time: CPU Time + time spent waiting (for disk, I/O, ...).
CPU time: Proportional to Instruction Count

Q. Once ISA is set, who can influence instruction count?
A. Compiler writer, application developer.

Q. Static count? (lines of program printout)
Q. Or dynamic count? (trace of execution)
A. Dynamic.

Rationale: Every additional instruction you execute takes time.

\[
\frac{\text{CPU time}}{\text{Program}} \propto \frac{\text{Machine Instructions}}{\text{Program}}
\]

Q. How does a architect influence the number of machine instructions needed to run an algorithm?
A. Create new instructions: instruction set architect.
Q. How can architects (not technologists) reduce clock period?

Q. What ultimately limits an architect's ability to reduce clock period?

We will revisit these questions later in lecture ...

Rationale:
We measure each instruction's execution time in "number of cycles". By shortening the period for each cycle, we shorten execution time.
Completing the performance equation

What factors make different programs have different CPIs?

- Cache behavior varies.
- Instruction mix varies.
- Branch prediction varies.

We need all three terms, and only these terms, to compute CPU Time!

Q. When is it OK to compare clock rates?

A. When other RHS terms are equal.
Consider Lecture 2 single-cycle CPU ...

All instructions take 1 cycle to execute every time they run.

CPI of any program running on machine? 1.0

"average CPI for the program" is a more-useful concept for more complicated machines ...
Recall Lecture 2: Multi-flow VLIW CPU

Q. Which right-hand-side term decreases with “N”? 

<table>
<thead>
<tr>
<th>Seconds Program</th>
<th>Instructions Program</th>
<th>Cycles Instruction</th>
<th>Seconds Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. This one gets smaller.</td>
<td></td>
<td></td>
<td>A. We hope this one doesn’t grow.</td>
</tr>
</tbody>
</table>

Syntax: ADD $8 $9 $10  Semantics:$8 = $9 + $10

<table>
<thead>
<tr>
<th>opcode</th>
<th>rs</th>
<th>rt</th>
<th>rd</th>
<th>shamt</th>
<th>funct</th>
</tr>
</thead>
</table>

Syntax: ADD $7 $8 $9  Semantics:$7 = $8 + $9

N x 32-bit VLIW yields factor of N speedup!

Multiflow: N = 7, 14, or 28 (3 CPUs in product family)
Consider machine with a data cache...

A program’s load instructions “stride” through every memory address.

The cache never “hits”, so every load goes to DRAM (100x slower than loads that go to cache).

Thus, the average number of cycles for load instructions is higher for this program.

Thus, the average number of cycles for all instructions is higher for this program.

\[
\frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \quad \frac{\text{Cycles}}{\text{Instruction}} \quad \frac{\text{Seconds}}{\text{Cycle}}
\]

Thus, program takes longer to run!
CPI as an analytical tool to guide design

Machine CPI
(throughput, not latency)

Multiply 5
Other ALU 1
Load 2
Store 2
Branch 2

5 x 30 + 1 x 20 + 2 x 20 + 2 x 10 + 2 x 20
100
= 2.7 cycles/instruction

Now we know how to optimize the design...

Program Instruction Mix

Multiply 30%
Branch 20%
Load 20%
Store 10%
Other ALU 20%

Where program spends its time
Final thoughts: Performance Equation

\[
\frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}
\]

Goal is to optimize execution time, not individual equation terms.

- Machines are optimized with respect to program workloads.
- The CPI of the program. Reflects the program’s instruction mix.
- Clock period. Optimize jointly with machine CPI.
Invented the “one ISA, many implementations” business model.
Amdahl’s Law (of Diminishing Returns)

If enhancement “E” makes multiply infinitely fast, but other instructions are unchanged, what is the maximum speedup “S”? 

Where program spends its time

\[ S = \frac{1}{(\text{post-enhancement } \%) / 100\%} = \frac{1}{48\%/100\%} = 2.08 \]

Attributed to Gene Amdahl -- “Amdahl’s Law”

What is the lesson of Amdahl’s Law?

Must enhance computers in a balanced way!
Amdahl’s Law in Action

The program spends 30% of its time running code that can not be recoded to run in parallel.

\[ S = \frac{1}{(30\% + \frac{70\%}{N}) / 100\%} \]

<table>
<thead>
<tr>
<th>CPUs</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>(\infty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speedup</td>
<td>1.54</td>
<td>1.85</td>
<td>2.1</td>
<td>2.3</td>
<td>3.3</td>
</tr>
</tbody>
</table>
Real-world 2006: 2 CPUs vs 4 CPUs

20 in iMac
Core Duo 2, 2.16 GHz
$1500

Mac Pro
2 Dual-Core Xeons, 2.66 GHz
$3200 w/ 20 inch display.
## Real-world 2006: 2 CPUs vs 4 CPUs

**2 cores on one die.**

**4 cores on two dies.**


<table>
<thead>
<tr>
<th></th>
<th>Speedmark 4.5</th>
<th>Adobe Photoshop CS2</th>
<th>Cinema 4D XL 9.5.21</th>
<th>Compressor 2.1</th>
<th>iMovie 6.0.1</th>
<th>iTunes 6.0.4</th>
<th>Unreal Tournament 2004</th>
<th>Finder</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OVERALL SCORE</strong></td>
<td><strong>SUITE</strong></td>
<td><strong>RENDER</strong></td>
<td><strong>MPEG2 Encode</strong></td>
<td><strong>AGED EFFECT</strong></td>
<td><strong>MP3 ENCODE</strong></td>
<td><strong>FRAME RATE</strong></td>
<td><strong>ZIP ARCHIVE</strong></td>
<td></td>
</tr>
<tr>
<td>20-inch iMac Core 2 Duo/2.16GHz</td>
<td>245</td>
<td>1:55</td>
<td>1:01</td>
<td>2:37</td>
<td>0:52</td>
<td>1:03</td>
<td>74.4</td>
<td>2:22</td>
</tr>
<tr>
<td>Mac Pro 2.66GHz (Standard)</td>
<td>299</td>
<td>1:25</td>
<td>0:28</td>
<td>1:47</td>
<td>0:38</td>
<td>0:48</td>
<td>91.3</td>
<td>2:01</td>
</tr>
</tbody>
</table>

**Caveat:** Mac Pro CPUs are server-class and have architectural advantages (better I/O, ECC DRAM, etc).

### Simple audio and video tasks: easier to parallelize.

### ZIPing a file: very difficult to parallelize.

### Source: Macworld
Break
Timing
CPU time: Proportional to Clock Period

Q. How can architects (not technologists) reduce clock period?

Q. What ultimately limits an architect’s ability to reduce clock period?

In this part of lecture: we answer these questions ...

Time
Program \( \propto \) Time
One Clock Period

Rationale:
We measure each instruction’s execution time in “number of cycles”.
By shortening the period for each cycle, we shorten execution time.
Goal: Determine minimum clock period
A Logic Circuit Primer

“Models should be as simple as possible, but no simpler ...” — Albert Einstein.
Inverters: A simple transistor model

Inverter

\[
\begin{array}{c|c|c}
\text{In} & \text{Out} \\
\hline
0 & 1 \\
1 & 0 \\
\end{array}
\]

\[
\text{Out} = \overline{\text{In}}
\]

Correctly predicts logic output for simple static CMOS circuits.

Circuit

\[
\begin{array}{c|c|c}
\text{In} & \text{Out} \\
\hline
0 & 1 \\
1 & 0 \\
\end{array}
\]

pFET. A switch. “On” if gate is grounded.

nFET. A switch. “On” if gate is at Vdd.

Extensions to model subtler circuit families, or to predict timing, have not worked well ...
Transistors as water valves. (Cartoon physics)

If electrons are water molecules, transistor strengths (W/L) are pipe diameters, and capacitors are buckets ...

A “on” p-FET fills up the capacitor with charge.

A “on” n-FET empties the bucket.

This model is often good enough ...
What is the bucket? A gate’s “fan-out”.

“Fan-out”: The number of gate inputs driven by a gate’s output.

- Driving other gates slows a gate down.
- Driving wires slows a gate down.
- Driving it’s own parasitics slows a gate down.
A closer look at fan-out ...

Each logic cell contributes capacitance.

Driving more gates adds delay.

Linear model works for reasonable fan-out.

FO4: Fanout of four delay.

Delay time of an inverter driving 4 inverters.
Propagation delay graphs ...

- Cascaded gates:

Gate Delay
- Cascaded gates:

INVERS
TRANSFER
FUNCTION

INVERS
TRANSFER
FUNCTION
Worst-case delay through combinational logic

T2 might be the worst-case delay path (critical path)

\[ x = g(a, b, c, d, e, f) \]

If \( d \) going 0-to-1 switches \( x \) 0-to-1, delay is \( T1 \).
If \( a \) going 0-to-1 switches \( x \) 0-to-1, delay is \( T2 \).
It would be surprising if \( T1 > T2 \).
Why “might”? Wires have delay too ...

- Wires possess distributed resistance and capacitance

- Time constant associated with distributed RC is proportional to the square of the length

- Signals are typically “rebuffered” to reduce delay:
Clocked Logic Circuits
From Delay Models to Timing Analysis

Timing Analysis

What is the smallest T that produces correct operation?

<table>
<thead>
<tr>
<th>f</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 MHz</td>
<td>1 µs</td>
</tr>
<tr>
<td>10 MHz</td>
<td>100 ns</td>
</tr>
<tr>
<td>100 MHz</td>
<td>10 ns</td>
</tr>
<tr>
<td>1 GHz</td>
<td>1 ns</td>
</tr>
</tbody>
</table>
Timing Analysis and Logic Delay

Register:
An Array of Flip-Flops

If our clock period \( T > \) worst-case delay through CL, does this ensure correct operation?
Flip Flops have internal delays ...

Value of D is sampled on positive clock edge. Q outputs sampled value for rest of cycle.

t_setup

t_clk-to-Q
Flip-Flop delays eat into "time budget"

\[ T \geq \tau_{\text{clk} \rightarrow Q} + \tau_{\text{CL}} + \tau_{\text{setup}} \]
Clock skew also eats into “time budget”

As $T \to 0$, which circuit fails first?

$T \geq T_{CL} + T_{\text{setup}} + T_{\text{clk} \to Q} + \text{worst case skew.}$
Clocks have dedicated wires (low skew)

"Clock tree"

Flip flop clock inputs are the "leaves" of the "tree".

From: Xilinx Spartan 3 data sheet. Virtex is similar.
Gold wires form clock tree.
Clock Tree Delays, IBM “Power” CPU
Figure 7

Clock Tree Delays, IBM Power
Some Flip Flops have “hold” time ...

Does flip-flop hold time affect operation of this circuit? Under what conditions?

What is the intended function of this circuit?

t_clk-to-Q + t_inv > t_hold
For correct operation.
Searching for processor critical path

RegFile
rs1
rs2
ws
rd1
rd2
RegDest
RegWr
Ext
ExtOp
ALUsrc
ALUctr
Data Memory
Addr
Din
Dout
MemWr
MemToReg

Control Lines

Equal

Combinational Logic

Clk
Addr
Data
InstrMem

op
rs
rt
immediate
Searching for processor critical path

Timing Analysis

What is the smallest T that produces correct operation?

Must consider all connected register pairs.

Q. Why might I suspect this one?
A. Very long wire on the path.
Combinational paths for IBM Power 4 CPU

The critical path

Most wires have hundreds of picoseconds to spare.

Timing Estimation

Predicting a processor’s clock rate early in the project

Timing Closure

Meeting (or exceeding!) the timing estimate

Floorplanning: Essential to meet timing.

(Intel XScale 80200)
Q. How can architects (not technologists) reduce clock period?

A. Shorten the machine's critical path.

Q. What ultimately limits an architect’s ability to reduce clock period?

A. Clock-to-Q, setup times, 2-D floorplanning geometry.

Rationale:
We measure each instruction’s execution time in “number of cycles”. By shortening the period for each cycle, we shorten execution time.

\[
\frac{\text{Time}}{\text{Program}} \propto \frac{\text{Time}}{\text{One Clock Period}}
\]
On Thursday

Pipeline design - with enough detail to do a design.

<table>
<thead>
<tr>
<th>Th 1/30</th>
<th>Pipelining</th>
<th>Appendix C.2-6.</th>
</tr>
</thead>
</table>

Have fun in section!