MYTHBUSTING MODERN HARDWARE TO GAIN “MECHANICAL SYMPATHY”

Martin Thompson
@MJPT777
Myth - 1

“CPUs are not getting faster”
Myth 1 – “CPUs Are Not Getting Faster”

• “The Free Lunch Is Over” – Herb Sutter
  > The issue is clock speeds cannot continue to get faster.
  > However clock speeds are not everything!

• Let’s word split of the “Alice in Wonderland” text

<table>
<thead>
<tr>
<th>Processor</th>
<th>Model</th>
<th>Operations/sec</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Core 2 Duo</td>
<td>CPU P8600 @ 2.40GHz</td>
<td>1434</td>
<td>(2008)</td>
</tr>
<tr>
<td>Intel Xeon</td>
<td>CPU E5620 @ 2.40GHz</td>
<td>1768</td>
<td>(2010)</td>
</tr>
<tr>
<td>Intel Core</td>
<td>CPU i7-2677M @ 1.80GHz</td>
<td>2202</td>
<td>(2011)</td>
</tr>
<tr>
<td>Intel Core</td>
<td>CPU i7-2720QM @ 2.20GHz</td>
<td>2674</td>
<td>(2011)</td>
</tr>
</tbody>
</table>
1-per-cycle 256-bit multiply, add, and shuffle
Load double the data with Intel microarchitecture (Sandy Bridge) !!!

* Not fully pipelined
Myth 1 – “CPUs Are Not Getting Faster”

Nehalem 2.8GHz

$ perf stat <program>

6975.000345 task-clock # 1.166 CPUs utilized
   2,065 context-switches # 0.296 K/sec
      126 CPU-migrations # 0.018 K/sec
      14,348 page-faults # 0.002 M/sec
22,952,576,506 cycles # 3.291 GHz
   7,035,973,150 stalled-cycles-frontend # 30.65% frontend cycles idle
  8,778,857,971 stalled-cycles-backend # 38.25% backend cycles idle
35,420,228,726 instructions # 1.54 insns per cycle
                # 0.25 stalled cycles per insn
  6,793,566,368 branches # 973.988 M/sec
 285,888,040 branch-misses # 4.21% of all branches

5.981211788 seconds time elapsed
Myth 1 – “CPUs Are Not Getting Faster”

Sandy Bridge 2.4GHz

$ perf stat <program>

5888.817958 task-clock # 1.180 CPUs utilized
  2,091 context-switches # 0.355 K/sec
  211 CPU-migrations # 0.036 K/sec
  14,148 page-faults # 0.002 M/sec
19,026,773,297 cycles # 3.231 GHz
  5,117,688,998 stalled-cycles-frontend # 26.90% frontend cycles idle
  4,006,936,100 stalled-cycles-backend # 21.06% backend cycles idle
35,396,514,536 instructions # 1.86 insns per cycle
                          # 0.14 stalled cycles per insn
  6,793,131,675 branches # 1153.565 M/sec
  186,362,065 branch-misses # 2.74% of all branches

4.988868680 seconds time elapsed
Myth - 1

“CPUs are not getting faster”
Myth - 2

“Memory Provides Random Access”
Myth 2 – “Memory Provides Random Access”

• What do we mean by “Random Access”?  
  > Should it not really be “Arbitrary Access”?  
  > Ideally we would like $O(1)$ latency, where 1 is small
Memory Ordering

Core 1
- Registers
- Execution Units
- MOB
- LF/WC Buffers
- L1
- L2
- L3

Core 2
- Registers
- Execution Units
- MOB
- LF/WC Buffers
- L1
- L2
- L3
- Load Buffer
- Store Buffer

Core n
- Registers
- Execution Units
- MOB
- LF/WC Buffers
- L1
- L2
- L3
Cache Structure & Coherence

L1(D) - 32K
L1(I) - 32K
L2 - 256K
L3 - 8-20MB

LF/WC Buffers
MOB
128 bits

L0(I) – 1.5k µops
16 Bytes

256 bits
128 bits

32 Bytes

64-byte “Cache-lines”
TLB
Pre-fetchers
SRAM
MESI+F State Model
QPI
Memory Controller
System Agent
TLB
Pre-fetchers
Ring Bus
QPI Bus
Memory Channels
Main Memory

Bank Select, Pre-charge + RAS + CAS

Columns

Memory Array 4096 * 1024 * 16

Ranks are Banks in parallel

Memory Controller

Channel

Write Buffer

Row Buffer

Memory Module

Bank 0
Bank 1
Bank n

DRAM
Myth 2 – “Memory Provides Random Access”

• “The real design action is in the memory sub-systems – caches, buses, bandwidth, and latency.” – Richard Sites (DEC Alpha Architect)
  > No point making faster CPUs when we cannot feed them fast enough

• Let’s look at the latencies measured by the SiSoftware tool
  > Intel i7-3960X (Sandy Bridge E)

<table>
<thead>
<tr>
<th></th>
<th>L1D</th>
<th>L2</th>
<th>L3</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>3 clocks</td>
<td>11 clocks</td>
<td>14 clocks</td>
<td>6.0 ns</td>
</tr>
<tr>
<td>In-Page Random</td>
<td>3 clocks</td>
<td>11 clocks</td>
<td>18 clocks</td>
<td>22.0 ns</td>
</tr>
<tr>
<td>Full Random</td>
<td>3 clocks</td>
<td>11 clocks</td>
<td>38 clocks</td>
<td>65.8 ns</td>
</tr>
</tbody>
</table>
Myth - 2

“Memory Provides Random Access” Busted
Myth - 3

“HDDs Provide Random Access”
Myth 3 – “HDDs Provide Random Access”

Sectors 512/4096 Bytes

Command Queue
Read/Write Cache + Pre-fetcher

Zone Bit Recording (ZBR)

Sequential Read Speed
(higher is better)
Myth 3 – “HDDs Provide Random Access”

What Makes up an IO operation?

- **Command Overhead**
  > Time for the electronics to process and schedule the request – Sub millisecond

- **Seek Time**
  > Time to move the read/write arm to the appropriate cylinder
  > Seek and Settle – 0-6ms Server Drive, 0-15ms Laptop Drive

- **Rotational Latency**
  > For a 10K RPM disk a rotation takes 6ms so average will be 3ms

- **Data Transfer**
  > Dependent on media and interface transfer speeds – 100-200 MB/s
Myth 3 – “HDDs Provide Random Access”

Are there tricks to hide latency and increase IOPs?

• Dual Actuators/Arms
  - Half the seek time at increased expense

• Multiple Copies of Data
  - Cut rotational delay at reduced drive capacity and increased write cost

• Command Queues
  - Apply elevator algorithms to smooth out latency which work well

• Battery/Capacitor backed Cache
  - Store up commands to handle burst traffic but not sufficient for sustained load
Myth - 3

"HDDs Provide Random Access"

Busted
Myth - 4

“SSDs Provide Random Access”
Myth 3 – “SSDs Provide Random Access”

Logical 2MB Block

256/512 Cells

4096/8192 Cells

Row == Page 4KB

Read/Write Pages

- File A
- File B
- File C
- Deleted
- Free Space

MLC / SLC Cells

Deleted means Garbage Collection TRIM?

Erase Block!!!
Myth 3 – “SSDs Provide Random Access”

Clean

After fill and torture

Intel 320 SSD

AnandTech Performance Tests

Read

Write

Beware Write Amplification!
Myth 3 – “SSDs Provide Random Access”

• Random re-writes hurt performance and wear out the drive
  > Block erase is 2ms!

• Reads have great random and sequential performance

• Append only writes have great random and sequential performance

<table>
<thead>
<tr>
<th>@40K IOPs</th>
<th>Average (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read 4K Random</td>
<td>0.1 - 0.2</td>
<td>2 - 30</td>
</tr>
<tr>
<td>Write 4K Random</td>
<td>0.1 - 0.3</td>
<td>2 - 500</td>
</tr>
</tbody>
</table>
Myth - 4

“SSDs Provide Plausible Random Access”
Questions?


Twitter: [@mjpt777](https://twitter.com/mjpt777)