i4 vs. i2 – A Real World Performance Comparison

or

„The Proof of the Pudding is in the Eating...“

Thilo Lauer
Technical Consultant IV
HP ASC & PSC DACH
Hewlett Packard GmbH

Dr. Martin P.J. Zinser
Senior Expert Trading IT
Eurex Frankfurt AG
Agenda

Talk split in 2 parts:

– The VMS perspective (Thilo)
– The application perspective (Martin)

• Motivation
  – Existing benchmarks are, well, just benchmarks
  – Translate “memory latency” to the CIO...
  – Recent large i4 Blade rollout

• Customer intro
  – Company, Business, Trading
  – IT landscape, VMS Cluster setup
  – Move to i4 & VSI VMS V8.4-1H1

• Performance observations
Motivation

• To compare systems, you can...
  – Trust in shiny data sheets from vendors
    Problem: no one is that naive, right?
  – Evaluate existing benchmarks
    Problem: isolated testing, match results to apps
  – Test and measure your own application(s)
    Problem: create comparable test environment

• Recent i4 Upgrade of large VMS cluster
  – Stepwise replacement of i2 systems
  – System latency measured in µsec.
  – Application hand-tuned over many years
  – Good example for procedures, complex environment, high perf. requirements
  – Good example of smoothness of HW&OS upgrade
Customer Introduction

DBAG:

• Founded in 1990
• 2-site VMS Cluster since Day 1
• One of the largest Exchange organizations in the world
• ... more from Martin
Performance observations - General

• General problem: no real baseline
  – Benchmarks use X iterations to measure Y counts of <something>
  – Application load is not reproduceable

• Approaches:
  – Use SPL tracing (and look at the nsec counters)
  – Measure via direct correlation
  – Manually create ratios of dependent entities
  – Delegate investigation to Apps maintainers

• Expectation:
  – Fact: going from 1.73 to 2.53 GHz (+46 %)
  – See improvement > 0
  – approach benchmark results, maybe see above clock speed increase
Benchmark results

### i2 vs i4 Memory Bandwidth

<table>
<thead>
<tr>
<th></th>
<th>MB/Sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL860c i2</td>
<td>2968</td>
</tr>
<tr>
<td>BL860c i4</td>
<td>5045</td>
</tr>
<tr>
<td>RX2800 i2</td>
<td>2070</td>
</tr>
<tr>
<td>RX2800 i4</td>
<td>4832</td>
</tr>
</tbody>
</table>

- **BL860c i2 (1.47ghz)**
- **BL860c i4 (2.4ghz)**
- **RX2800 i2 (1.73ghz)**
- **RX2800 i4 (2.67ghz)**

**Relative Bandwidth Performance:**
- **BL860c i2**: 1.00
- **BL860c i4**: 1.33
- **RX2800 i2**: 1.14
- **RX2800 i4**: 1.57

**Changes:**
- +70%
- +133%

### i2 vs i4 Memory Latency

<table>
<thead>
<tr>
<th></th>
<th>Latency (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL860c i2</td>
<td>237</td>
</tr>
<tr>
<td>BL860c i4</td>
<td>196</td>
</tr>
<tr>
<td>RX2800 i2</td>
<td>243</td>
</tr>
<tr>
<td>RX2800 i4</td>
<td>108</td>
</tr>
</tbody>
</table>

- **BL860c i2 (1.47ghz)**
- **BL860c i4 (2.4ghz)**
- **RX2800 i2 (1.73ghz)**
- **RX2800 i4 (2.67ghz)**

**Changes:**
- -17%
- -56%

### i2 vs i4 Floating Point Performance

<table>
<thead>
<tr>
<th></th>
<th>Relative Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL860c i2</td>
<td>1.00</td>
</tr>
<tr>
<td>BL860c i4</td>
<td>1.33</td>
</tr>
<tr>
<td>RX2800 i2</td>
<td>1.14</td>
</tr>
<tr>
<td>RX2800 i4</td>
<td>1.57</td>
</tr>
</tbody>
</table>

- **BL860c i2 (1.47ghz)**
- **BL860c i4 (2.4ghz)**
- **RX2800 i2 (1.73ghz)**
- **RX2800 i4 (2.67ghz)**

**Relative Performance:**
- +33%
- +38%

### i2 vs i4 Integer Performance

<table>
<thead>
<tr>
<th></th>
<th>Relative Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL860c i2</td>
<td>1.00</td>
</tr>
<tr>
<td>BL860c i4</td>
<td>1.26</td>
</tr>
<tr>
<td>RX2800 i2</td>
<td>1.17</td>
</tr>
<tr>
<td>RX2800 i4</td>
<td>1.49</td>
</tr>
</tbody>
</table>

- **BL860c i2 (1.47ghz)**
- **BL860c i4 (2.4ghz)**
- **RX2800 i2 (1.73ghz)**
- **RX2800 i4 (2.67ghz)**

**Relative Performance:**
- +26%
- +27%
Application Characteristics

Before:
14 idle CPUs
Application Characteristics

After:
29 idle CPUs
Results – T4 File Sizes

I4 Systems produce 30% bigger T4 data files!
## Results – Spinlock hold time

@SYS$EXAMPLES:SPL

<table>
<thead>
<tr>
<th>Spinlock</th>
<th>Caller's PC</th>
<th>% Time Held</th>
<th>Acquires /sec</th>
<th>Average (nsec)</th>
<th>Spinwaits /sec</th>
<th>Average Spinwait</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCHED</td>
<td></td>
<td>5.7</td>
<td>18151.9</td>
<td>3136</td>
<td>8679.4</td>
<td>8092</td>
</tr>
<tr>
<td>8071B900 PROCESS_MANAGEMENT+4D000</td>
<td>[SCHED_ROUTINES+01C20 / SCH$IDLE+01220]</td>
<td>2.3</td>
<td>11174.2</td>
<td>2021</td>
<td>7721.2</td>
<td>7060</td>
</tr>
<tr>
<td>807E2570 PROCESS_MANAGEMENT+00113C70</td>
<td>[AST_QUEUEING+01BE0 / SCH_STD$QAST+019A0]</td>
<td>1.9</td>
<td>7098.9</td>
<td>2646</td>
<td>3254.8</td>
<td>6369</td>
</tr>
<tr>
<td>807B2270 EXE$WFLAND_COMMON_C+003B0</td>
<td>[SYSWAIT+003B0 / EXE$WFLAND_COMMON+003B0]</td>
<td>1.6</td>
<td>9560.6</td>
<td>1675</td>
<td>4400.3</td>
<td>5993</td>
</tr>
<tr>
<td>807AC170 SCH$POSTEF_C+00150</td>
<td>[POSTEF+01330 / SCH$POSTEF+00150]</td>
<td>1.6</td>
<td>9560.6</td>
<td>1675</td>
<td>4400.3</td>
<td>5993</td>
</tr>
<tr>
<td>80786ED0 EXE$HIBER_INT_C+006C0</td>
<td>[SYSPCNTRL+04130 / EXE$HIBER_INT+006C0]</td>
<td>1.1</td>
<td>3866.1</td>
<td>2814</td>
<td>1773.3</td>
<td>5367</td>
</tr>
</tbody>
</table>

Comparison of Spinlocks held from 10 identical routines:

Improvement between **13 %** and **63 %**
Results – Disk Ops vs. CPUx Int mode

CPU 6 = preferred CPU for FG devices

Disk I/Os over CPU 6 Int mode answers „How many I/Os does my system perform with a given CPU load?“

Average of 4 i2 systems = 377

\[ f(x) = \frac{10000}{26x} = 385x \]
Results – Disk Ops vs. CPUx Int mode

CPU 6 = preferred CPU for FG devices

Disk I/Os over CPU 6 Int mode answers „How many I/Os does my system perform with a given CPU load?“

Average of 5 i4 systems = 597

Result: 597/377 = + 58 %
Results – FCP Calls vs. FCP CPU

\[ f(x) = \frac{2000}{25x} = 80x \]

\[ f(x) = \frac{2000}{17x} = 118x \]

Result: \( \frac{80}{118} = +47\% \)
Results – FCP Calls vs. FCP CPU

+31 %  
+37 %  
+41 %
Results – Locking vs. CPUx Int Mode

Average over 2 * 8 measurements: 83

Average over 2 * 8 measurements: 208

Improvement 151 %
Results – TCP TxPkts vs. CPUx Int Mode

Average over 8 measurements: 327

Average over 8 measurements: 595

Improvement 82 %
Results – MPSynch vs. User Mode

Observation:
- OS overhead scales linearly with # of CPUs
- Improvement of **22 %** (**33 %** with KERNEL mode)
Questions?

thilo.lauer@hpe.com
thilo.lauer@hp.com
thilo.lauer@compaq.com
thilo.lauer@digital.com