Dual Processor Performance Characterization
for XML Application-Oriented Networking

Jianxun Jason Ding and Abdul Waheed
Cisco Systems, Inc.
{jding,abwaheed}@cisco.com

Abstract

There is a growing trend to insert application intelligence into network devices. Processors in this type of Application-Oriented Networking (AON) devices are required to handle both packet-level network I/O intensive operations as well as XML message-level CPU intensive operations. In this paper, we investigate performance effect of dual processing via (1) Hyperthreading, (2) uni-processor to dual-processor, and (3) single-core to dual-core, on both packet-level and XML message-level traffic. We analyze and cross-examine the dual processing effect from both high-level performance as well as processor microarchitectural perspectives. We employ on-chip performance counters to measure cycles per instruction, cache misses, bus utilization, and branch miss predictions for this work. Our results show a significant improvement in dual-core Pentium M processor over Hyperthreaded Xeon processor for AON workload. These results will not only provide insight to processor designers, but also help architects of AON devices to select from alternative processors with restrictions to use one or two physical CPUs due to space and power consumption limitations.

1. Introduction

Traditional network devices were designed to conduct OSI L1 to L4 operations. With the advance of both hardware and software technologies, network devices are able to process larger number of tasks per unit of time. Growing number of network devices are expanding their operations from packet-level L1-L4 networking to message-level L5-L7 application-oriented networking (AON) [3].

Web operations employ EXtensible Markup Language (XML) technology intensively, including message oriented operations on AON devices. Selecting the right processor unit(s) for AON type of network device to process both L1-L4 network operations and L5-L7 application oriented operations is no simple task. From the processing power angle, system architects would like to know the extent of performance improvement that one can expect by adding symmetric processor units. In case of network devices, adding processing units is often limited by available space on system boards as well as power consumption budgets. Therefore, a majority of current networking device designs can accommodate only one to two physical processing units on a single system board. Thus, we limit the scope of our study from single to dual processing units for this paper. From the workload spectrum angle, architects also need to know the performance characteristics and scaling factor when operations change from packet-level networking oriented processing to XML message-level application oriented processing. This paper is an attempt to address these issues holistically by presenting empirical processor level performance data, their analyses, and comparison across processor macroarchitectures and AON workloads.

A number of microarchitecture level performance studies examine the symmetric multiprocessing impact of enterprise application workloads. For instance, Padma Apparao et al. provided microarchitecture level scaling characteristics of XML workload on Xeon processors with and without Hyperthreading [1]. Other studies have considered architectural characteristics of TCP network traffic routing and forwarding workloads [6][7]. To the best of our knowledge, there is a lack of any in-depth microarchitecture level performance study of dual processing impact of a mixed network traffic and XML application level processing workload. Since AON workloads fall in this mixed network I/O and CPU-intensive category, our work is an effort to fill this void. Our primary objective is to lend insight to AON device architects. Additionally, this study provides feedback to processor architects as AON workloads increasingly become common in the enterprise.

In this paper, we will use both packet based L1-L4 network workloads and message based L5-L7 workloads to examine the performance effect of dual processing via (1) Hyperthreading technology; (2) uni-processor to dual-processor; and (3) single-core to
dual-core. For this purpose, we selected two platforms: one based on multiple Intel Xeon processors with Hyperthreading support and another based on an Intel Pentium Mobile dual-core processor. We employ a custom-built application, which provides two layers of functionality instead of using any monolithic application to flexibly move between two extremes of network I/O and CPU intensive workload categories. The base level of this application provides HTTP message proxying support. The second level provides support for executing various XML functions on message content, which are requested through HTTP POST messages. Our performance analyses will distinguish across these two layers of functionality. We use on-chip processor performance counters to collect event occurrences. We analyze and correlate these low levels of measurements to characterize the dual processing effect of our application with respect to the above two layers of functionality.

We hope that the results presented in this paper will: (1) help AON designers choose the appropriate processors to process both packet-level networking traffic and XML message-level application oriented traffic; (2) let microprocessor architects be aware of symmetric processing effects from both network I/O and XML based application oriented workloads; and (3) spawn more research efforts in the application oriented networking area.

In Section 2, we present an overview of on-chip performance counters based performance work. We describe our evaluation methodology, tools, and performance metrics in Section 3. We present measurement based results using on-chip performance counters in Sections 4 and 5 with in-depth analyses of Pentium M and Xeon architectures and dual processing with respect to packet-level and XML message-level AON workloads. We conclude in Section 6 with a summary of contributions of this work.

2. Related Work

On-chip performance counters [2][11][12] have been used for performance evaluation and tuning of parallel applications. For example, there have been studies of using performance counters on IBM SP2 distributed memory multicomputer system [11] and SGI Origin2000 distributed shared memory multiprocessor system for tuning and characterization of Computational Fluid Dynamics applications [9]. Most of these efforts are primarily focused on application tuning and using processor level measurements as a tool to facilitate the process of identifying bottlenecks in application code. System architects cannot use these studies to understand the interaction of a specific type of workload with processor microarchitecture to help them in designing systems.

There are a few studies that have focused on microarchitecture level performance characteristics of selected workload. Pentium Pro performance counters were used to characterize the SPEC benchmarks with respect to processor microarchitecture [2]. Apparao et al. used performance counters on Xeon processor to analyze the characteristics of XML based server workloads with respect to Hyperthreading and multiprocessor configurations [1]. This study is primarily focused on Xeon processor. While our study is also focused on XML based application workloads, it has a wider scope as our workloads are network I/O intensive as well as CPU intensive. Such workloads are typical in service oriented network devices that process XML contents, which arrive through TCP/IP based message passing. In addition, we have the opportunity to compare and contrast the microarchitectures of two processors: Intel Pentium M and Intel Xeon. This wider scope is essential to allow the system architects to use the results reported in this study.

3. Analysis Methodology

3.1. Processors and Systems under Test

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Pentium M</th>
<th>Xeon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CPUs</td>
<td>1 core and 2 cores</td>
<td>1 CPU and 2 CPUs</td>
</tr>
<tr>
<td>Hyperthreading</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>CPU Speed</td>
<td>1.83GHz</td>
<td>3.16GHz</td>
</tr>
<tr>
<td>L1 D Cache</td>
<td>32KB</td>
<td>16KB</td>
</tr>
<tr>
<td>L1 I Cache</td>
<td>32KB</td>
<td>12KB trace cache</td>
</tr>
<tr>
<td>L2 Cache</td>
<td>2MB</td>
<td>1MB</td>
</tr>
<tr>
<td>Frontside Bus</td>
<td>667MHz</td>
<td>667 MHz</td>
</tr>
<tr>
<td>DRAM Size</td>
<td>2GB</td>
<td>2GB</td>
</tr>
<tr>
<td>OS Version</td>
<td>RHAS4 2.6 Kernel</td>
<td>RHAS4 2.6 Kernel</td>
</tr>
<tr>
<td>Compiler</td>
<td>gcc 3.4.5 –O3</td>
<td>gcc 3.4.5 –O3</td>
</tr>
</tbody>
</table>

We use two Intel processors based systems for this investigation. One of them is based on a dual-core Intel Pentium M processor while the other is an SMP system consisting of two Intel Xeon processors with Hyperthreading capabilities. Specific details about each of these systems are presented in Table 1.

In this paper, we evaluate two combinations of dual-core Pentium M based system using either a
single core or both the cores. In case of Xeon based system, we evaluate three combinations: single logical CPU, two logical CPUs (due to Hyperthreading), and two physical CPUs (without Hyperthreading). We provide the notations that we shall use for reporting results for these cases in Table 2 for the sake of consistency.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1CPm</td>
<td>Pentium M processor booted with SMP Linux kernel using only one of two cores with maxcpus=1 bootloader flag</td>
</tr>
<tr>
<td></td>
<td>2CPm</td>
<td>Pentium M processor booted with SMP Linux kernel using both the cores with maxcpus=2</td>
</tr>
<tr>
<td></td>
<td>1LPx</td>
<td>Xeon processor with Hyperthreading disabled from BIOS and booted with SMP Linux kernel using a single CPU with maxcpus=1</td>
</tr>
<tr>
<td></td>
<td>2LPx</td>
<td>Xeon processor with Hyperthreading enabled from BIOS and booted with SMP Linux kernel using two logical CPUs with maxcpus=2</td>
</tr>
<tr>
<td></td>
<td>2PPx</td>
<td>Xeon processors with Hyperthereading disabled from BIOS and booted with SMP Linux kernel using two physical CPUs with maxcpus=2</td>
</tr>
</tbody>
</table>

3.2. Workload Characterization

Our primary workload mimics networked XML message content processing applications. Therefore, the workload consists of two types of operations: XML content processing related computation and network I/O operations. XML content processing is different from traditional computation intensive applications. Traditional computation intensive applications stem from scientific and engineering problems that exercise floating point processing and memory-to-memory copying capabilities of a processor. In the context of XML content processing, computation mainly involves character string manipulation functions, such as copying, concatenation, parsing, tokenization, and matching. These operations rarely involve any floating point computation. However, they exercise logical operations, data and instruction cache architecture, branch prediction, and memory-to-memory transfer bandwidth. Network I/O involves some (non floating point) compute operations on packet headers and moving payload bits among memory, CPU, and PCI based network interface hardware.

We use two sources to generate XML message content processing and network I/O workload: a custom-developed XML server applications and a bulk data transfer based benchmark. Details of these workload sources follow.

3.2.1. XML Server Application

A large number of commercial applications leverage multiple processors (or cores) through multi-threaded programming paradigm. This server application uses POSIX threads to utilize multiple CPUs or cores for processing and data transfer. XML server application consists of multiple threads, which are kept equal to the number of (logical) CPUs that the operating system can detect to leverage from dual CPUs or cores. Since AON focuses on XML message (or packet) processing functions, our experimental server application implements two basic XML functions: XPath evaluation and schema validation. In addition to these XML functions, this experimental server application also works as an HTTP proxy to provide a baseline to measure the performance of other CPU intensive XML processing functions.

We focus on three use cases of XML server application in terms of processing incoming XML request through HTTP POST messages: (1) HTTP request forwarding, (2) XML content based routing, and (3) XML schema validation.

HTTP Forward Request (FR) represents the simplest infrastructure level web service—proxying for HTTP requests. This use case represents the overhead of processing a message to forward to a default destination. Unless there are configuration or execution errors in the server application system under test, this case should result in the highest throughput compared to all other use cases because it does not involve any content processing. It simply exercises the network I/O components of the application and the system. Thus, this use case is valuable to validate the baseline for the test harness as well as the system under test. FR uses the same XML message that will be used for CBR.

Content Based Routing (CBR) performs three operations on the incoming XML message content: (a) content lookup; (b) content routing based on a pattern match or lack thereof; and (c) forwarding of message to selected destination endpoint. Content lookup is performed using XPath evaluation. In this case, we use an XML message with SOAP envelope containing a \(<quantity>\) element. Using XPath expression \(//quantity/text()\), XML server application determines whether this tag is found and is equal to string “1”. If this condition is evaluated to true, the message is forwarded to the intended endpoint; otherwise, it is
forwarded to a default error handling endpoint. There are filler text elements to increase the overall message size to conform to the AONBench specified 5 Kbytes message size [6]. This specification uses only one Xpath. The goal of using this case is to provide a mix of CPU intensive and network I/O intensive mixed use case. Following SV use case is a predominant CPU intensive case.

**Schema Validation** (SV) use case allows the server application to compare the incoming XML message to a pre-stored schema to determine whether it conforms to the specified grammar. If the message turns out to be valid according to the schema, it is sent to one destination; otherwise, it is sent to the error endpoint. A modified input message can verify whether the XML server application is executing this use case correctly.

The objective of this case is to measure the performance during schema validation process. This is a CPU intensive task and its complexity will depend on the incoming XML message and the schema.

We use netperf benchmarks in two different modes: (1) end-to-end mode and (2) loopback mode. In end-to-end mode, netserver and netperf processes run on separate systems connected through a Gigabit network. This is standard TCP stream benchmark mode to measure end-to-end network bandwidth for TCP bulk data transfer and represents an extreme network I/O case. In the loopback mode, both netserver and netperf processes run on the same physical host. In this case, TCP stream test exercises the local CPU and is used to represent an extreme CPU intensive case.

Our workload characterization appropriately covers two dimensions of processor performance: CPU intensive and network I/O intensive operations performance. As show in Figure 1, XML server application falls in the space defined by these two workload dimensions. We use Netperf in end-to-end and loopback modes to establish a baseline along each of these two dimensions.

### 3.3. Profiling Tool and Performance Metrics

We use Intel VTune Performance Analyzer [5] tool to access on-chip performance counters for measurements presented in this paper. VTune is a profiling tool for Intel processors based systems running Linux or Windows operating systems. It can provide a callgraph as well as sampling based profiling for an application or entire system, respectively. Both types of profiling approaches employ on-chip counters to accurately measure processor level events. Some of the processor level events, which are relevant to application performance include: clock ticks, instructions retired, L1 cache misses, L2 cache misses, data memory accesses, branch instruction retired, branch mispredictions retired, and TLB misses. Ratios of multiple basic processor level events can be used to derive higher performance metrics.

We use sampling based VTune profiling to get a global picture of processor utilization for both system and application level activities. We also use traffic throughput as our high level XML server application throughput in an end-to-end measurement setup. Instead of directly using throughput values, we use throughput scaling values, which represent ratio of throughput of the system with respect to single processing unit based throughput. Higher value of this metric is desirable. Our microarchitecture level performance metrics include: Cycles per Instructions Retired (CPI), L2 cache misses per retired instruction (L2MPI), bus transactions per retired instruction (BTPI), and branch mispredictions per retired branch instruction ratio (BrMPR). Lower values of these...
metrics are desirable except for bus accesses per retired instruction.

4. Baseline Measurements and Analysis

Figure 2 presents the Netperf throughput in Mbits per second for both modes on all five processor configurations. We note the following performance characteristics for both modes:

1) End-to-end mode Netperf throughput is limited by the network bandwidth of 1 Gbps. Considering TCP/IP stack overhead, a TCP application cannot utilize all of the available network bandwidth. A TCP application that achieves greater than 90% of available network. Thus, all processor configurations are capable of saturating the Gigabit Ethernet bandwidth with end-to-end bulk data communication.

2) Netperf in loopback mode achieves the highest throughput on Pentium M with single core (1CPm). This is closely followed by the throughput of a single Xeon processor without Hyperthreading (1LPx). For both Pentium M and Xeon processors, throughput degrades from single to dual processing units. Surprisingly, this impact is more severe for Xeon (2PPx) compared to Pentium M (2CPm). On Pentium M, one expects a degradation due to shared L2 cache between two cores whereas two Xeon processors do not share any caches. We shall explain this behaviour through performance counter based measurements later in this section. Another noteworthy point is that single Xeon processor throughput with Hyperthreading enabled (2LPx) incurs only slight degradation compared to 1LPx.

Table 3. Performance Metrics for Netperf benchmark in loopback and end-to-end modes.

<table>
<thead>
<tr>
<th>Workload</th>
<th>Metrics</th>
<th>1CPm</th>
<th>2CPm</th>
<th>1LPx</th>
<th>2LPx</th>
<th>2PPx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netperf-loopback</td>
<td>CPI</td>
<td>3.03</td>
<td>6.05</td>
<td>6.38</td>
<td>7.70</td>
<td>22.13</td>
</tr>
<tr>
<td></td>
<td>L2MPI</td>
<td>0.00</td>
<td>0.35</td>
<td>0.00</td>
<td>23.32</td>
<td>24.64</td>
</tr>
<tr>
<td></td>
<td>Bus transactions per retired inst. (%)</td>
<td>0.00</td>
<td>9.84</td>
<td>0.19</td>
<td>0.10</td>
<td>10.48</td>
</tr>
<tr>
<td></td>
<td>Branch instructions per retired inst. (%)</td>
<td>36</td>
<td>34</td>
<td>18</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>BrMPR (%)</td>
<td>0.96</td>
<td>0.70</td>
<td>3.23</td>
<td>3.04</td>
<td>2.30</td>
</tr>
<tr>
<td>Netperf</td>
<td>CPI</td>
<td>3.46</td>
<td>6.27</td>
<td>8.10</td>
<td>18.52</td>
<td>11.53</td>
</tr>
<tr>
<td></td>
<td>L2MPI</td>
<td>0.05</td>
<td>0.08</td>
<td>0.33</td>
<td>2.89</td>
<td>2.71</td>
</tr>
<tr>
<td></td>
<td>Bus transactions per retired inst. (%)</td>
<td>2.13</td>
<td>5.99</td>
<td>0.53</td>
<td>0.95</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>Branch instructions per retired inst. (%)</td>
<td>33</td>
<td>34</td>
<td>18</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>BrMPR (%)</td>
<td>0.85</td>
<td>0.83</td>
<td>1.68</td>
<td>3.96</td>
<td>1.87</td>
</tr>
</tbody>
</table>

We resort to using on-chip performance counters to analyze the throughput of Netperf benchmark in two modes to provide an appropriate baseline for more complex AON workloads. We collect all relevant event counts using VTune during Netperf tests and calculate microarchitecture level performance metrics. Among other metrics, we also present the ratio of branch instructions retired to the overall instructions retired for
all processor configurations. We note that this ratio is constant within Pentium M and Xeon processor based configurations. However, this ratio is double for Pentium M configurations compared to Xeon based configurations. We shall defer the discussion of this difference until Section 5.5.

We arrive at the following conclusions related to each metric presented in Table 3 while comparing it across processor configurations:

1) CPIs increase from single to dual processor unit configurations for same type of processors. For end-to-end mode, this is due to second processor unit spending most of its cycles idle as netperf is the only active process. In case of loopback mode, other processor unit runs the netserver process. In this case, larger CPI is a consequence of resource related stalls. These resources could be L2 cache (for 2CPm) or bus (for both 2CPm and 2PPx).

2) L2MPI is small for Pentium M for both Netperf configurations. However, it is larger for Xeon dual processing units (2LPx and 2PPx) for both end-to-end and loopback modes. This appears to be a result of smaller L2 cache size for Xeons compared to Pentium M. Sharing of L2 cache for 2LPx is a major cause of throughput degradation in loopback mode compared to 1LPx configuration.

3) Bus traffic volume increases by an order of magnitude from single to dual (physical) processor configuration for Netperf in loopback mode for both Pentium M and Xeon processors. Larger bus traffic results in increased conflicts for bus accesses, which mean larger number of stall cycles. Same behaviour can be observed for Netperf in end-to-end mode but due to smaller volumes of data transfer, this trait is not equally pronounced.

4) BrMPR is expected to be similar within Pentium M and Xeon processors for single to dual processor configurations but can differ across two processors due to the differences in their microarchitectures. This is true in end-to-end mode (except 2LPx) because only netperf process is active. However, the picture becomes murky for loopback mode because BrMPR is based on contributions from both netperf and netserver processes.

There are some key points that we learn from this baseline analysis and need to utilize them while characterizing AON workloads in the following section. For network I/O intensive workloads, L2 cache and bus traffic can limit the scalability from single to dual processing units. CPI, branch misprediction ratio, and branch instruction frequencies relative to instructions retired provide a valuable tool to examine the performance of CPU intensive workloads.

5. XML Server Application Measurements and Analysis

In the following five subsections, we analyze performance counters based measurements of AON workloads from four angles: (1) performance trend from network I/O intensive workloads to CPU intensive workloads; (2) comparison of Pentium M and Xeon processors using the performance counters under discussion; (3) dual processing effect within the Pentium M and Xeon processor families; and (4) comparison of dual processing effect across the Pentium M and Xeon processor families.

5.1. End-to-End Throughput

Figure 3 shows the dual processor scaling for the three use cases. According to Figure 3, from FR to CBR to SV, the percentage of network I/O operations decreases and the proportion of computation operations grows. We observe that the single to dual Pentium M processor scaling increases from 1.51 to 1.84 and then to 1.91 for FR, CBR, and SV, respectively. Scalability enhances due to larger number of CPU cycles that can be utilized from the two physical processor cores as workload becomes increasingly CPU intensive. The dual processing effect of 1LPx to 2LPx is interesting: it shows the reverse trend, from 1.49, 1.32, to 1.12 for FR, CBR, and SV, respectively. This trend indicates that one physical Xeon processor with Hyperthreading enabled performs better for running the network I/O intensive workload than running the CPU intensive workload. With larger number of network I/O
operations, the Xeon execution unit pipelines will have more empty slots, leaving more room for parallel processing; while the CPU intensive workload will fill up more pipeline slots, leaving less room for processing other computation threads. In Figure 3, we also observe that 1LPx to 2PPx (one logical Xeon processor to two physical Xeon processors) scales quite well in all three cases: FR, CBR and SV. It even scales better than the 1CPm to 2CPm (one to two core Pentium M) situation. This is because the two physical cores in the 2CPm case share the same L2 cache, while each of the two physical Xeon processors has its own dedicated L2 cache.

5.2. Cycles per Instruction (CPI)

Using the same workload, CPI can be used to compare processor execution efficiency with respect to their microarchitecture. Table 4 presents the CPI measurements of AON workloads on five selected system under test configurations.

<table>
<thead>
<tr>
<th>Workload</th>
<th>1CPm</th>
<th>2CPm</th>
<th>1LPx</th>
<th>2LPx</th>
<th>2PPx</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV</td>
<td>1.02</td>
<td>1.05</td>
<td>1.91</td>
<td>3.50</td>
<td>1.96</td>
</tr>
<tr>
<td>CBR</td>
<td>1.12</td>
<td>1.22</td>
<td>2.26</td>
<td>4.34</td>
<td>2.32</td>
</tr>
<tr>
<td>FR</td>
<td>2.24</td>
<td>2.96</td>
<td>5.71</td>
<td>7.65</td>
<td>5.92</td>
</tr>
</tbody>
</table>

First, CPI increases as workload varies from CPU intensive to network I/O intensive on all processor configurations. This is because I/O operations spend some empty cycles waiting for I/O operations to finish, regardless of how efficient the I/O operation modules are designed and coded. The CPIs of Pentium M processors are significantly lower than Xeon’s CPIs. This difference in CPIs indicates that the Netburst microarchitecture of Xeon processor is less efficient compared to Pentium M processor when executing this I/O intensive workload, even with its higher CPU frequency and the Hyperthreading technology. The CPIs for the 2CPm case are quite similar to the 1CPm case for SV, CBR, and FR workloads due to proportional increase of aggregate counts of Clocktics and Instructions Retired events on two cores. The CPIs increase significantly from 1LPx to 2LPx because the same physical execution units cannot really utilize all (double) of the processing cycles with the Hyperthreading technology by proportionally increasing the instruction retired. When the workload changes from network I/O intensive (FR) to CPU intensive (SV), the CPI upsurge damps down, as more pipeline stages can utilize the Hyperthreading technology. The CPIs of 1LPx vs. 2PPx are almost the same, similar to the trend of 1CPm to 2CPm, because of the availability of two sets of physical processors.

5.3. L2 Cache Misses

In this subsection, we try to discern the throughput and CPI based scaling and performance characteristics of FR, CBR, and SV workloads with respect to the cache architectures of Pentium M and Xeon processors. Figure 4 compares the L2 cache misses per retired instruction (L2MPI).

We observe and analyze following points from the L2MPI comparisons of Figure 4:

1) With growing network I/O operations intensity, L2MPI values also increase. This is due to the nature of data involved in network I/O operations, which consist of packets based message payload. Such data do not have any temporal re-use and simply have to incur cache misses to move to or from the memory. Increasing CPU intensive workload cases exhibit proportionally growing temporal re-use characteristics with decreasing L2MPI values.

2) The Pentium M processor has smaller L2MPIs compared to those of the Xeon processor for both single and dual processing unit cases. This is because (a) the Pentium M processor core has a larger L2 cache (2 MB L2 in Pentium M vs. 1 MB L2 in Xeon); and (b) the Pentium M L1 data cache size is twice the size of the Xeon L1 data cache (32 KB vs. 16 KB).

3) The L2MPIs increase from 1CPm to 2CPm because L2 cache in 2CPm is shared by the two processor cores. The dual processor units in the 2LPx case are two logical units as opposed to two
physical units for 2CPm case. While a comparatively larger number of instructions finish execution for 2LPx, the L2 cache misses do not increase proportionally. This is mainly due to increased re-use of instructions in L2 cache by two threads, which execute same sequence of operations. Thus we see L2MPIs decrease from 1LPx to 2LPx. On the other hand, each of the dual processors in 2PPx has its own dedicated L2 cache. Since there is little dependency between the instruction flows in our workloads, the L2MPIs remain the same from 1PPx to 2PPx.

4) L2MPI exhibit same characteristics for 1LPx to 2LPx as for 1CPm to 2CPm single to dual processing unit transitions. Only difference is the larger magnitude of L2MPI for Xeon compared to Pentium M, which is explains by 2) above.

5) L2 cache miss characteristics provide insight into the throughput and CPI trends with respect to workloads and microarchitectures. As in case of CPI, Pentium M microarchitecture results in lower L2MPI for AON workloads compared to Xeon.

### 5.4 Bus Transactions per Instruction

Memory performance is an extension of cache performance for any workload because memory references that miss L1 and L2 caches result in accesses to the main memory. This is particularly true for network I/O intensive workloads where caches do not enhance temporal locality but simply hide latency through spatial locality. In this subsection, we examine the memory bus traffic.

Figure 5 presents the bus transactions per (retired) instruction (BTPI) for three AON workloads. We can glean following information from this plot:

1) BTPI increase from CPU intensive workloads to network I/O intensive workloads in all the processors configurations. This is because the network I/O operations need to move data around via the CPU-memory bus. Thus the network I/O intensive workloads have larger BTPI compared to CPU intensive operations.

2) The L2 cache in Pentium M processor is twice the capacity of the Xeon L2 cache. So one would expect 1CPm BTPIs to be roughly one half of the corresponding 1LPx BTPIs. But Figure 5 shows that 1CPm and 1LPx BTPI values are in the same proximity. This is because Pentium M implements the Intel Smart Memory Access [10] technology with a new capability called “memory disambiguation” with two advanced prefetchers for L2 Cache. As a result, Pentium M processors execute more prefetches and additional reloads to L2 Cache. These additional memory to L2 loads cause the BTPI increase, offsetting the BTPI reduction by the larger L2 cache.

3) The BTPIs also increase from 1CPm to 2CPm because both cores share the same L2 cache in the dual core Pentium M configuration. For Xeon processor, the dual processing effect on bus utilization from 1LPx to 2PPx is insignificant due to two independent processors with independent L2 caches. With approximately double the number of accesses as well as instructions retired, the L2MPI ratio remains same for 1LPx to 2PPx.

4) BTPI for 2CPm is higher compared to 2PPx owing to increased number prefetches to implement memory disambiguation to support speculative instruction execution in Pentium M microarchitecture.

![Figure 5. Bus transactions per retired instruction (as %) for AON AML use cases.](image)

Bus transactions based measurements are consistent with overall throughput, CPI, and L2MPI for selected AON workloads. In addition, these measurements agree with the expected behavior with respect to processor microarchitectures as well as AON workload characteristics.

### 5.5 Branch Misprediction Ratios

Branch prediction is an important feature in contemporary processors to reap maximum benefit from speculative execution and pipelining. Inefficient branch prediction results in excessive stall cycles, higher CPI ratios, and lower application performance. Importance of branch prediction is directly tied to the frequency of branch instructions within a workload.

Table 5 presents the branch frequency as a ratio of branch instructions retired to the total number of instructions retired for each of the three AON workloads (as a percentage). While both Pentium M and Xeon based systems execute the same application...
code, there are significant differences in branch frequencies in the overall instruction mix. It appears that Pentium M retires close to double the number of branch instructions relative to overall instructions compared to Xeon for three AON workloads. This appears to be a consequence of “Wide Dynamic Execution” based microarchitecture that simultaneously fetches and executes four instructions at a time [10]. More branch instructions are speculatively executed per instruction retired in Pentium M compared to Xeon. Also, network I/O intensive workload (FR) has about 25% more branch instructions compared to CPU intensive workloads (CBR and SV).

**Table 5. Branch instruction retired per instruction retired (Branch Frequency)**

<table>
<thead>
<tr>
<th>Workload</th>
<th>1CPm</th>
<th>2CPm</th>
<th>1LPx</th>
<th>2LPx</th>
<th>2PPx</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV</td>
<td>27</td>
<td>28</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>CBR</td>
<td>28</td>
<td>27</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>FR</td>
<td>35</td>
<td>36</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
</tbody>
</table>

We measured the Branch Misprediction Ratios (BrMPR) for both Pentium M and Xeon processors based cases. BrMPR measurements are presented in Table 6.

**Table 6. Branch misprediction ratios**

<table>
<thead>
<tr>
<th>Workload</th>
<th>1CPm</th>
<th>2CPm</th>
<th>1LPx</th>
<th>2LPx</th>
<th>2PPx</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV</td>
<td>1.98</td>
<td>1.97</td>
<td>3.62</td>
<td>4.61</td>
<td>3.65</td>
</tr>
<tr>
<td>CBR</td>
<td>1.07</td>
<td>1.04</td>
<td>2.01</td>
<td>2.91</td>
<td>1.96</td>
</tr>
<tr>
<td>FR</td>
<td>1.13</td>
<td>1.21</td>
<td>2.65</td>
<td>3.96</td>
<td>2.71</td>
</tr>
</tbody>
</table>

We observe following points in BrMPR measurements:

1) The CPU intensive workload SV has higher branch misprediction ratios than relatively more I/O intensive workloads (FR and CBR) on both Pentium M and Xeon processors.

2) The BrMPR of Pentium M processor is significantly lower than Xeon processor BrMPR for all three workloads. This difference in the BrMPR of two processors indicates that branch prediction mechanism implemented in the Pentium M processor is more efficient than that of Xeon.

3) BrMPR remains largely unaffected by the number of processing units: 1CPm to 2CPm as well as 1LPx to 2PPx. However, one exception is the 2LPx case where enabling Hyperthreading with one physical Xeon processor results in at least 25% increase in BrMPR compared to 1LPx based cases. These measurements uncover a negative correlation between Hyperthreading and BrMPR on Xeon. Clearly, there is no such correlation when we consider 1LPx to 2PPx case.

4) BrMPR for 2PPx is almost double compared to 2CPm. This is consistent with the microarchitecture differences between Pentium M and Xeon, which are relevant and applicable for dual processing units as well.

5) BrMPR difference between Pentium M and Xeon is a significant factor, which translates into higher throughput and lower CPIs for Pentium M running AON workloads. These measurements also expose the unfavorable impact of Hyperthreading for BrMPR on Xeon for the same workloads.

**6. Summary**

We have provided an in-depth performance evaluation of two leading Intel processors with respect to XML based application oriented networking workloads. These workloads represent a growingly important application for general-purpose processors and it is increasingly important for system architects to understand the performance trade-offs of choosing one type of processor over the other for such applications in single and dual processor configurations. At the same time, processor architects also need to receive feedback on the suitability of a particular microarchitecture for XML based network I/O and CPU intensive application workloads. We have strived to address these issues in this study.

The main findings of this work can be summarized as:

1) Hyperthreading technology in Xeon processor scales poorly compared to dual Xeon or dual core Pentium M processors’ scaling for AON workloads. Our analysis identifies higher CPI, lower bus traffic related to memory accesses, and higher branch misprediction ratios for Xeon with Hyperthreading enabled due to sharing of same physical processor between two logical instruction streams.

2) Dual core Pentium M based system provides balanced scaling for AON workloads for network I/O intensive as well as CPU intensive mixes. Using on-chip counters based measurements of processor level events, we relate these characteristics to improved Pentium M microarchitecture with wide dynamic execution, smart memory access technologies, and efficient branch prediction mechanism.

3) Comparing throughput scaling characteristics of dual core Pentium M to dual Xeon processors (without Hyperthreading) for AON workloads, we
observed slightly lower scaling factors for network I/O intensive workloads for Pentium M. We explain this behaviour due to the conflicts for shared L2 cache by two Pentium M cores as well as increased memory traffic due to Smart Memory Access technology. However, this difference diminishes for CPU intensive cases due to improved temporal locality of data, which undergo XML content based processing.

4) Overall, Pentium M microarchitecture excels Xeon Netburst architecture in terms of superior CPI and branch misprediction rates. Pentium M microarchitecture provides balanced performance for mixed network and CPU intensive workloads, which are prevalent in XML based application oriented network services.

5) We unexpectedly uncovered some interesting characteristics of Xeon and Pentium M processor microarchitectures for branch instructions and mispredictions. For same workloads, Pentium M retires twice the number of branch instructions over those retired by Xeon with respect to the total number of retired instructions. This appears to be related to wider instruction fetches and execution in Pentium M to support speculative execution. However, this strategy pays off in terms of lower CPI and lower branch misprediction ratios for AON workloads.

6) We also discovered that a single Xeon processor with Hyperthreading enabled incurs significantly higher branch mispredictions compared to single or dual Xeon processors without Hyperthreading for the same AON workloads. Our speculation is that it is related to some dependence of branch prediction on sharing of physical resources among two logical instruction streams. This overhead shows up in terms of higher CPI as well as lower throughput for AON workloads.

Future work includes extending this study to multi-core based AON devices. We also plan to investigate performance characteristics on crucial AON operations such as deep packet inspection, XML parsing, and crypto functions.

7. References


